A robust nonsmooth generalized-α scheme for flexible systems with impacts

Alejandro Cosimo[1,2], Javier Galvez[1], Federico J. Cavalieri[2], Alberto Cardona[2] and Olivier Brüls[1]

[1]University of Liège, Department of Aerospace and Mechanical Engineering, Allée de la Découverte 9, 4000 Liège, Belgium

[2]Centro de Investigación de Métodos Computacionales (CIMEC), Colectora Ruta Nac Nro 168, Km 0, Paraje El Pozo, 3000 Santa Fe, Argentina

Abstract

The aim of this work is the development of a robust and accurate time integrator for the simulation of the dynamics of multibody systems composed by rigid and/or flexible bodies subject to frictionless contacts and impacts. The integrator is built upon a previously developed nonsmooth generalized-α scheme time integrator which was able to deal well with nonsmooth dynamical problems avoiding any constraint drift phenomena and capturing vibration effects without introducing too much numerical dissipation. However, when dealing with problems involving nonlinear bilateral constraints and/or flexible elements, it is necessary to adopt small time step sizes to ensure the convergence of the numerical scheme. In order to tackle these problems more efficiently, a fully decoupled version of the nonsmooth generalized-α method is proposed in this work, avoiding these inconveniences. Several examples are considered to assess its accuracy and robustness.

Keywords: nonsmooth contact dynamics, flexible multibody system, generalized-α method, time-stepping schemes

1 Introduction

The aim of this work is the development of methods for the numerical simulation of the nonsmooth dynamics of multibody systems involving rigid and/or flexible elements which can be subject to frictionless contacts and impacts. These systems are characterized by bilateral constraints which are associated to kinematic joints interconnecting the bodies, and by unilateral constraints stemming from the frictionless contacts and impacts. An additional difficulty comes from the presence of flexible elements, with vibration effects that need to be efficiently captured by the numerical scheme.

For the robust and accurate simulation of such systems, special attention must be paid to the adopted time integration scheme as it not only has to deal successfully with the nonsmooth character of the problem but also with the vibration effects. Time integrators for nonsmooth dynamics can be classified in two main groups: event-driven and time-stepping integrators. The former are based on the exact detection of impacts by accordingly adapting the time step

size. However, they become inefficient in situations involving a large number of impact events. A different strategy is adopted in this work, which falls under the category of time-stepping integrators. These techniques share the common feature that the time step size does not need to be adapted to impact events. The most widespread time-stepping integrators for nonsmooth dynamical systems are the Schatzman–Paoli scheme [1, 2], which is based on a central difference scheme, and the Moreau–Jean scheme [3, 4, 5], which is based on a θ-method. Despite their robustness for dealing with problems involving a large number of impacts, they generally lead to poor approximations of vibration phenomena. Additionally, in the Moreau–Jean scheme, the constraints are only imposed at velocity level which leads to the violation of the constraints at position level and, in consequence, a drift phenomenon is observed [6].

These problems have been studied in the literature of nonsmooth dynamics, mainly dealing with the use of higher order integrators for the smooth or free-flight part of the motion [6, 7, 8, 9, 10], and with the simultaneous imposition of the constraints at position and at velocity levels [11, 12, 6]. The imposition of the constraints at acceleration level was also analyzed recently by Brüls *et al.* [10]. The current work takes as starting point the nonsmooth generalized-α (NSGA) introduced by Brüls *et al.* in [6] and proposes a modification to improve its robustness for problems with nonlinear bilateral constraints and/or flexible components.

The NSGA deals with the transient simulation of nonsmooth dynamical systems comprised of rigid and/or flexible bodies, kinematic joints and frictionless contacts. It is characterized by the splitting of the involved fields into a smooth and a (nonsmooth) impulsive contribution, where the former is integrated with second order accuracy by means of the generalized-α scheme and the latter with first-order accuracy. Also, the involved unilateral and bilateral constraints are exactly satisfied both at position and at velocity levels. This results in a numerical scheme which involves three coupled subsets of equations or sub-problems to be solved at each time step: one for the smooth prediction of the motion, and two others for correcting that prediction at position and at velocity levels with the nonsmooth contributions. The existing coupling stems from the adopted splitting in which the smooth sub-problem depends on the position correction and on the velocity jump. Therefore, if a semi-smooth Newton approach is used to solve the derived formulation without making any additional assumption, at each nonlinear iteration the method would have to deal monolithically with the complete set of unknowns. In order to avoid this issue, Brüls *et al.* [6] proposed to neglect the terms coupling the smooth sub-problem with the other ones from the tangent matrix. The advantage of this procedure is that the algorithm can be described as a sequence of three sub-problems, instead of having to solve the complete set of equations monolithically. This approximation is fully justified when the adopted step size tends to zero. However, for problems with flexible bodies and nonlinear bilateral constraints, this approximation led to a slow convergence of the global scheme, or even to the divergence of the scheme if a small enough step size was not adopted. In order to overcome this difficulty, the current work proposes to modify the way to do the splitting in order to ensure a full decoupling of the different subsets of equations, so that the three sub-problems can be processed in a sequential decoupled manner without any approximation. Here, the adjective *decoupled* is used because the resulting numerical scheme involves the sequential solution of three sub-problems *without neglecting any term* in the discrete problem. This implies that the proposed decoupled scheme results in a robust alternative especially for problems characterized by nonlinear bilateral constraints and flexible elements. In this work, only frictionless problems are considered, but the procedure can be easily extended to friction problems as well.

The paper is organized as follows. In Section 2, the nonsmooth equations of motion are stated. Several alternatives that can be found in the literature for the splitting of the smooth and nonsmooth components of the motion are addressed in Section 3 in order to subsequently

introduce a splitting which leads to a decoupled formulation of the integrator. Next, the numerical performance of the proposed method is studied in Section 4. The conclusions and future work are given in Section 5.

2 Equations of motion

After spatial semi-discretization, the equations of motion for a frictionless multibody system with unilateral and bilateral constraints expressed at velocity level are written in the following form:

$$\begin{aligned} \dot{\boldsymbol{q}}^+ &= \boldsymbol{v}^+ && \text{(1a)} \\ \boldsymbol{M}(\boldsymbol{q})\,\mathrm{d}\boldsymbol{v} - \boldsymbol{g}_{\boldsymbol{q}}^T\,\mathrm{d}\boldsymbol{i} &= \boldsymbol{f}(\boldsymbol{q},\boldsymbol{v},t)\,\mathrm{d}t && \text{(1b)} \\ -\boldsymbol{g}_{\boldsymbol{q}}^{\overline{\mathcal{U}}}\,\boldsymbol{v}^+ &= \boldsymbol{0} && \text{(1c)} \\ \text{if } g^j(\boldsymbol{q}) \le 0 \text{ then } 0 \le \boldsymbol{g}_{\boldsymbol{q}}^j\,\boldsymbol{v}^+ + e^j\,\boldsymbol{g}_{\boldsymbol{q}}^j\,\boldsymbol{v}^- &\perp \mathrm{d}i^j \ge 0, \quad \forall\, j \in \mathcal{U} && \text{(1d)} \end{aligned}$$

where

- t is time, and $\mathrm{d}t$ is the corresponding standard Lebesgue measure.
- $\boldsymbol{q}(t)$ is the vector of coordinates, which are absolutely continuous in time.
- $\mathcal{U}$ denotes the set of indices of the unilateral constraints, $\overline{\mathcal{U}}$ is its complementarity set, *i.e.*, the set of bilateral constraints, $\mathcal{C} = \mathcal{U} \cup \overline{\mathcal{U}}$ is the full set of constraints.
- $\boldsymbol{g}$ is the combined set of bilateral and unilateral constraints, and $\boldsymbol{g}_{\boldsymbol{q}}(\boldsymbol{q})$ is the corresponding matrix of constraint gradients.
- $\dot{\boldsymbol{q}}^+(t) = \lim_{\tau\to t,\tau>t} \dot{\boldsymbol{q}}(\tau)$ and $\boldsymbol{v}^+(t) = \lim_{\tau\to t,\tau>t} \boldsymbol{v}(\tau)$ are the right limits of the velocity, which are functions of bounded variations. Similarly, $\boldsymbol{v}^-(t) = \lim_{\tau\to t,\tau<t} \boldsymbol{v}(\tau)$ is the left limit of the velocity. It is assumed, without loss of generality, that $\boldsymbol{v}^+$ and $\boldsymbol{v}^-$ are related at an impact event by the Newton impact law $g_{\boldsymbol{q}}^j \boldsymbol{v}^+(t) = -e^j\, g_{\boldsymbol{q}}^j \boldsymbol{v}^-(t)$, where e^j is the coefficient of restitution at the contact point $j \in \mathcal{U}$. In what follows, for simplicity $\boldsymbol{v}(t)$ and $\dot{\boldsymbol{q}}(t)$ will be used to denote $\boldsymbol{v}^+(t)$ and $\dot{\boldsymbol{q}}^+(t)$, respectively.
- $\boldsymbol{f}(\boldsymbol{q},\boldsymbol{v},t) = \boldsymbol{f}^{ext}(t) - \boldsymbol{f}^{cin}(\boldsymbol{q},\boldsymbol{v}) - \boldsymbol{f}^{damp}(\boldsymbol{q},\boldsymbol{v}) - \boldsymbol{f}^{int}(\boldsymbol{q})$ collects the external, complementary inertia, damping and internal forces.
- $\boldsymbol{M}(\boldsymbol{q})$ is the mass matrix which may, in general, depend on the coordinates.
- $\mathrm{d}\boldsymbol{v}$ is the differential measure associated with the velocity $\boldsymbol{v}$ assumed to be of bounded variations.
- $\mathrm{d}\boldsymbol{i}$ is the impulse measure of the unilateral contact reaction and the bilateral constraint forces.
- The measures $\mathrm{d}\boldsymbol{v}$ and $\mathrm{d}\boldsymbol{i}$ have the following decomposition:

$$\mathrm{d}\boldsymbol{v} = \dot{\boldsymbol{v}}\,\mathrm{d}t + \sum_i (\boldsymbol{v}(t_i) - \boldsymbol{v}^-(t_i))\,\delta_{t_i} \qquad (2)$$

$$\mathrm{d}\boldsymbol{i} = \boldsymbol{\lambda}\,\mathrm{d}t + \sum_i \boldsymbol{p}_i\,\delta_{t_i} \qquad (3)$$

where $\boldsymbol{\lambda}$ is the vector of smooth Lagrange multipliers associated with the Lebesgue measurable constraint forces, δ_{t_i} the Dirac delta supported at t_i, and $\boldsymbol{p}_i$ is the impulse producing the jump at the instant t_i.

3 Decoupled version of the NSGA method

The original NSGA method proposed by Brüls *et al.* [6] is characterized by three coupled subproblems: one for the smooth part of motion, and two others for the nonsmooth contributions at position and velocity levels. In order to avoid solving the three systems of equations monolithically, the terms coupling the smooth prediction to the corrections at position and at velocity levels are neglected. However, as it will be shown in the numerical examples, this can have serious consequences on the convergence of the method.

The derivation of the time integration scheme to be proposed here follows the same key steps as described in [6]. However, two differences should be highlighted. Firstly, the splitting is modified in order to ensure a full decoupling of the different subsets of equations. Secondly, the equations are formulated using an augmented Lagrangian approach which combines Lagrange multiplier and penalty terms. The advantage of using an augmented Lagrangian method is the presence of a penalty term which adds convexity to the objective function and improves the convergence of the Newton iteration far from the solution. This factor does not influence the accuracy of the computed solution [13].

3.1 Splitting strategy

The splitting of the variables aims at isolating the impulsive terms from the smooth contributions to the motion. Let us consider a time interval $(t_n, t_{n+1}]$, and let $\dot{\tilde{\boldsymbol{v}}}(t)$ be a function of bounded variations, which can be defined in several different ways as it will be discussed below. Then, the nonsmooth impulsive contribution to the motion $\mathrm{d}\boldsymbol{w}$ can be defined by decomposing the measure of the velocity $\mathrm{d}\boldsymbol{v}$ as

$$\mathrm{d}\boldsymbol{v} = \mathrm{d}\boldsymbol{w} + \dot{\tilde{\boldsymbol{v}}}\,\mathrm{d}t \tag{4}$$

where $\dot{\tilde{\boldsymbol{v}}}\,\mathrm{d}t$ is a purely diffuse measure. The smooth contribution to the velocity field, denoted by $\tilde{\boldsymbol{v}}(t)$, is computed by integration of $\dot{\tilde{\boldsymbol{v}}}(t)$ with the initial values $\tilde{\boldsymbol{v}}(t_n) = \boldsymbol{v}(t_n)$. On the other hand, the smooth contribution to the position, denoted as $\tilde{\boldsymbol{q}}(t)$, is computed by integration of $\dot{\tilde{\boldsymbol{q}}}(t) = \tilde{\boldsymbol{v}}(t)$ with the initial values $\tilde{\boldsymbol{q}}(t_n) = \boldsymbol{q}(t_n)$. By construction, the variables $\tilde{\boldsymbol{v}}(t)$ and $\tilde{\boldsymbol{q}}(t)$ are, respectively, absolutely continuous and C^1 in time and, in this sense, they only capture a smooth part of the motion.

The smooth part of the trajectory is obtained from the time integration of the acceleration variable $\dot{\tilde{\boldsymbol{v}}}$ by a second order method, whereas $\mathrm{d}\boldsymbol{w}$ is integrated using a first-order Euler implicit scheme. Hence, it is recommended to capture as much information as possible in $\dot{\tilde{\boldsymbol{v}}}$ to gain accuracy. Several alternatives exist for defining the smooth part of motion in the splitting, as described next.

- The optimal choice would be to take $\dot{\tilde{\boldsymbol{v}}} \triangleq \dot{\boldsymbol{v}}$, so that $\dot{\tilde{\boldsymbol{v}}}$ captures all diffuse contributions in the equation of motion. This alternative was investigated in [10] based on a formulation of the constraints at acceleration level, bringing certain advantages such as the elimination of spurious oscillations of the constraints after impact events.

- Some simplifications can be proposed to avoid the need of using acceleration constraints and to eliminate the Linear Complementarity Problem (LCP) in the definition of $\dot{\tilde{\boldsymbol{v}}}$.

For instance, Chen *et al.* [7] defined $\dot{\tilde{\boldsymbol{v}}}$ to satisfy the equations of motion at almost any time t (when there is no impact) without accounting for the bilateral and the unilateral constraints and associated forces.

- A third alternative is proposed in [6], in which a compromise between the option of taking $\dot{\tilde{\boldsymbol{v}}} \triangleq \dot{\boldsymbol{v}}$ and that of Chen *et al.* [7] is made by defining an initial value problem for the smooth contributions in which the bilateral constraints and forces are considered. In this case, the smooth contributions to the position $\tilde{\boldsymbol{q}}(t)$, the velocity $\tilde{\boldsymbol{v}}(t)$, the acceleration $\dot{\tilde{\boldsymbol{v}}}(t)$ and the Lagrange multiplier $\tilde{\boldsymbol{\lambda}}(t)$ satisfy

$$\begin{aligned} \dot{\tilde{\boldsymbol{q}}} &= \tilde{\boldsymbol{v}} && \text{(5a)} \\ \boldsymbol{M}(\boldsymbol{q})\,\dot{\tilde{\boldsymbol{v}}} - \boldsymbol{g}_{\boldsymbol{q}}^T\,\tilde{\boldsymbol{\lambda}} &= \boldsymbol{f}(\boldsymbol{q},\boldsymbol{v},t) && \text{(5b)} \\ -\boldsymbol{g}_{\boldsymbol{q}}^{\overline{\mathcal{U}}}(\boldsymbol{q})\,\tilde{\boldsymbol{v}} &= \boldsymbol{0} && \text{(5c)} \\ \tilde{\boldsymbol{\lambda}}^{\mathcal{U}} &= \boldsymbol{0} && \text{(5d)} \end{aligned}$$

  with the initial values $\tilde{\boldsymbol{v}}(t_n) = \boldsymbol{v}(t_n)$ and $\tilde{\boldsymbol{q}}(t_n) = \boldsymbol{q}(t_n)$. It is remarked that only bilateral constraints at velocity level are taken into account in this formulation and that the Lagrange multipliers of the unilateral constraints are set to zero.

- A fourth alternative could be obtained by setting the smooth acceleration variable $\dot{\tilde{\boldsymbol{v}}}$ to zero so that all the dynamics would be integrated using the Euler implicit method (with a loss of accuracy with respect to the other options).

- The first three alternatives for the definition of $\dot{\tilde{\boldsymbol{v}}}$ have the disadvantage that the resulting equations depend on the total position and velocity, that is, they depend not only on the smooth components of the position and the velocity, but also on the nonsmooth (impulsive) components. Therefore, these formulations are characterized by a smooth sub-problem which is coupled with the set of equations defining the nonsmooth contributions. In order to avoid this coupling, we propose to define the smooth sub-problem as the solution of the following initial value problem:

$$\begin{aligned} \dot{\tilde{\boldsymbol{q}}} &= \tilde{\boldsymbol{v}} && \text{(6a)} \\ \boldsymbol{M}(\tilde{\boldsymbol{q}})\,\dot{\tilde{\boldsymbol{v}}} - \boldsymbol{g}_{\tilde{\boldsymbol{q}}}^T\,\tilde{\boldsymbol{\lambda}} &= \boldsymbol{f}(\tilde{\boldsymbol{q}},\tilde{\boldsymbol{v}},t) && \text{(6b)} \\ -\boldsymbol{g}_{\tilde{\boldsymbol{q}}}^{\overline{\mathcal{U}}}(\tilde{\boldsymbol{q}})\,\tilde{\boldsymbol{v}} &= \boldsymbol{0} && \text{(6c)} \\ \tilde{\boldsymbol{\lambda}}^{\mathcal{U}} &= \boldsymbol{0} && \text{(6d)} \end{aligned}$$

  with the initial values $\tilde{\boldsymbol{v}}(t_n) = \boldsymbol{v}(t_n)$ and $\tilde{\boldsymbol{q}}(t_n) = \boldsymbol{q}(t_n)$, and where the matrix of constraint gradients $\boldsymbol{g}_{\tilde{\boldsymbol{q}}}(\tilde{\boldsymbol{q}})$ is computed in terms of $\tilde{\boldsymbol{q}}$ only. It should be observed that this fifth definition of $\dot{\tilde{\boldsymbol{v}}}$ only depends on the smooth contributions to the motion, $\tilde{\boldsymbol{q}}$ and $\tilde{\boldsymbol{v}}$, a property that naturally leads to a sequence of decoupled sub-problems.

An elimination of $\mathrm{d}\boldsymbol{v}$ and $\dot{\tilde{\boldsymbol{v}}}$ from Eqs. (1b,4,6b) yields the equations for the nonsmooth contributions

$$\boldsymbol{M}(\boldsymbol{q})\,\mathrm{d}\boldsymbol{w} - \boldsymbol{g}_{\boldsymbol{q}}^T\,[\mathrm{d}\boldsymbol{i} - \tilde{\boldsymbol{\lambda}}\,\mathrm{d}t] = \boldsymbol{f}^*(\boldsymbol{q},\boldsymbol{v},\tilde{\boldsymbol{q}},\tilde{\boldsymbol{v}},\dot{\tilde{\boldsymbol{v}}},t)\,\mathrm{d}t \tag{7}$$

together with the set of bilateral and unilateral constraints Eqs. (1c-1d), and where

$$\boldsymbol{f}^*(\boldsymbol{q},\boldsymbol{v},\tilde{\boldsymbol{q}},\tilde{\boldsymbol{v}},\dot{\tilde{\boldsymbol{v}}},t) = \boldsymbol{f}(\boldsymbol{q},\boldsymbol{v},t) - \boldsymbol{f}(\tilde{\boldsymbol{q}},\tilde{\boldsymbol{v}},t) + (\boldsymbol{g}_{\boldsymbol{q}}^T - \boldsymbol{g}_{\tilde{\boldsymbol{q}}}^T)\tilde{\boldsymbol{\lambda}} - (\boldsymbol{M}(\boldsymbol{q}) - \boldsymbol{M}(\tilde{\boldsymbol{q}}))\;\dot{\tilde{\boldsymbol{v}}} \tag{8}$$

3.2 Constraints at position and velocity levels

The constraints in the derived set of equations are imposed at velocity level only. Moreau's viability Lemma [14] implies that the exact solution of these equations also satisfies the constraints at position level. However, the numerical solution will not satisfy them at position level because of a drift phenomenon. Therefore, the equations of motion are reformulated as in [6] such that the unilateral and bilateral constraints appear both at position and velocity levels. This procedure is inspired by the index-reduction proposed by Gear, Gupta and Leimkuhler for bilaterally constrained mechanical systems [15]. By introducing an additional Lagrange multiplier $\boldsymbol{\mu}$, the velocity equation (1a) is relaxed:

$$\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}} = \boldsymbol{M}(\boldsymbol{q})\boldsymbol{v} + \boldsymbol{g}_{\boldsymbol{q}}^{T}\,\boldsymbol{\mu}$$

and by adding the constraints on position, the set of equations to be solved becomes

$$\begin{aligned}
\boldsymbol{M}(\tilde{\boldsymbol{q}})\,\dot{\tilde{\boldsymbol{v}}} - \boldsymbol{g}_{\tilde{\boldsymbol{q}}}^{\overline{\mathcal{U}},T}\,\tilde{\boldsymbol{\lambda}}^{\overline{\mathcal{U}}} &= \boldsymbol{f}(\tilde{\boldsymbol{q}},\tilde{\boldsymbol{v}},t) && \text{(9a)}\\
-\boldsymbol{g}_{\tilde{\boldsymbol{q}}}^{\overline{\mathcal{U}}}\,\tilde{\boldsymbol{v}} &= \boldsymbol{0} && \text{(9b)}\\
\tilde{\boldsymbol{\lambda}}^{\mathcal{U}} &= \boldsymbol{0} && \text{(9c)}\\
\mathrm{d}\boldsymbol{v} &= \mathrm{d}\boldsymbol{w} + \dot{\tilde{\boldsymbol{v}}}\,\mathrm{d}t && \text{(9d)}\\
\boldsymbol{M}(\boldsymbol{q})\,\dot{\boldsymbol{q}} - \boldsymbol{g}_{\boldsymbol{q}}^{T}\,\boldsymbol{\mu} &= \boldsymbol{M}(\boldsymbol{q})\,\boldsymbol{v} && \text{(9e)}\\
-\boldsymbol{g}^{\overline{\mathcal{U}}}(\boldsymbol{q}) &= \boldsymbol{0} && \text{(9f)}\\
\boldsymbol{0} \le \boldsymbol{g}^{\mathcal{U}}(\boldsymbol{q}) &\perp \boldsymbol{\mu} \ge \boldsymbol{0} && \text{(9g)}\\
\boldsymbol{M}(\boldsymbol{q})\,\mathrm{d}\boldsymbol{w} - \boldsymbol{g}_{\boldsymbol{q}}^{T}\,[\mathrm{d}\boldsymbol{i} - \tilde{\boldsymbol{\lambda}}\,\mathrm{d}t] &= \boldsymbol{f}^{*}(\boldsymbol{q},\boldsymbol{v},\tilde{\boldsymbol{q}},\tilde{\boldsymbol{v}},\dot{\tilde{\boldsymbol{v}}},t)\,\mathrm{d}t && \text{(9h)}\\
-\boldsymbol{g}_{\boldsymbol{q}}^{\overline{\mathcal{U}}}\boldsymbol{v} &= \boldsymbol{0} && \text{(9i)}\\
\text{if } g^{j}(\boldsymbol{q}) \le 0 \text{ then } 0 \le \boldsymbol{g}_{\boldsymbol{q}}^{j}\,\boldsymbol{v} + e^{j}\,\boldsymbol{g}_{\boldsymbol{q}}^{j}\,\boldsymbol{v}^{-} &\perp \mathrm{d}i^{j} \ge 0, \quad \forall j \in \mathcal{U} && \text{(9j)}
\end{aligned}$$

3.3 Time stepping scheme

3.3.1 Velocity jump and position correction variables

Time integration of the velocity measure $\mathrm{d}\boldsymbol{v}$, Eq. (4), over the time interval $(t_n, t]$ gives

$$\boldsymbol{v}(t) = \int_{(t_n,t]} \mathrm{d}\boldsymbol{w} + \tilde{\boldsymbol{v}}(t) = \boldsymbol{W}(t_n;t) + \tilde{\boldsymbol{v}}(t) \tag{10}$$

with the definition $\boldsymbol{W}(t_n;t) = \int_{(t_n,t]} \mathrm{d}\boldsymbol{w}$ and where the properties $\boldsymbol{v}(t_n) = \tilde{\boldsymbol{v}}(t_n)$ and $\boldsymbol{q}(t_n) = \tilde{\boldsymbol{q}}(t_n)$ were used. By construction, the nonsmooth variable $\boldsymbol{W}(t_n;t)$ captures all velocity jumps taking place in the interval $(t_n, t]$.

In order to obtain a similar decomposition of the position variable, it is first recalled that $\boldsymbol{v}(t)$ and $\dot{\boldsymbol{q}}(t)$ are not formally equivalent, where the variable $\dot{\boldsymbol{q}}(t)$ is related to the position variable $\boldsymbol{q}(t)$ by the relation

$$\int_{(t_n,t]} \dot{\boldsymbol{q}}(\tau)\mathrm{d}\tau = \boldsymbol{q}(t) - \boldsymbol{q}(t_n) \tag{11}$$

Time integration of the smooth component $\tilde{\boldsymbol{v}}(t)$ gives

$$\int_{(t_n,t]} \tilde{\boldsymbol{v}}(\tau)\mathrm{d}\tau = \tilde{\boldsymbol{q}}(t) - \boldsymbol{q}(t_n) \tag{12}$$

where the property $\tilde{\boldsymbol{q}}(t_n) = \boldsymbol{q}(t_n)$ was used. Subtracting Eq. (12) from Eq. (11) results in

$$\boldsymbol{q}(t) = \int_{(t_n,t]} [\dot{\boldsymbol{q}}(\tau) - \tilde{\boldsymbol{v}}(\tau)]\mathrm{d}\tau + \tilde{\boldsymbol{q}}(t) = \boldsymbol{U}(t_n;t) + \tilde{\boldsymbol{q}}(t) \tag{13}$$

with the definition of the position correction $\boldsymbol{U}(t_n;t) = \int_{(t_n,t]}[\dot{\boldsymbol{q}}(\tau) - \tilde{\boldsymbol{v}}(\tau)]\mathrm{d}\tau$. Considering that $\boldsymbol{q}(t)$ is absolutely continuous and that $\tilde{\boldsymbol{q}}(t)$ is C^1, Eq. (13) implies that $\boldsymbol{U}(t_n;t)$ is absolutely continuous on the interval $(t_n, t_{n+1}]$.

In summary, we have the following expressions for the splitting of the total velocity and position

$$\boldsymbol{v}(t) = \tilde{\boldsymbol{v}}(t) + \boldsymbol{W}(t_n;t) \tag{14a}$$
$$\boldsymbol{q}(t) = \tilde{\boldsymbol{q}}(t) + \boldsymbol{U}(t_n;t) \tag{14b}$$

where the nonsmooth contributions $\boldsymbol{W}(t_n;t)$ and $\boldsymbol{U}(t_n;t)$ are defined by

$$\boldsymbol{W}(t_n;t) = \int_{(t_n,t]} \mathrm{d}\boldsymbol{w} \tag{15a}$$
$$\boldsymbol{U}(t_n;t) = \int_{(t_n,t]} [\dot{\boldsymbol{q}}(\tau) - \tilde{\boldsymbol{v}}(\tau)]\mathrm{d}\tau \tag{15b}$$

where, by construction, $\boldsymbol{W}(t_n;t_n) = \boldsymbol{0}$ and $\boldsymbol{U}(t_n;t_n) = \boldsymbol{0}$. In what follows, the multipliers $\boldsymbol{\Lambda}(t_n;t)$ and $\boldsymbol{\nu}(t_n;t)$ are taken as

$$\boldsymbol{\Lambda}(t_n;t) = \int_{(t_n,t]} [\mathrm{d}\boldsymbol{i} - \tilde{\boldsymbol{\lambda}}(\tau)\,\mathrm{d}\tau], \tag{16a}$$
$$\boldsymbol{\nu}(t_n;t) = \int_{(t_n,t]} [\boldsymbol{\mu}(\tau) + \boldsymbol{\Lambda}(t_n;\tau)]\,\mathrm{d}\tau, \tag{16b}$$

with $\boldsymbol{\Lambda}(t_n;t_n) = \boldsymbol{\nu}(t_n;t_n) = \boldsymbol{0}$.

3.3.2 Discrete approximations of W and U

The discrete approximations of the involved variables are now introduced. The time instant at which a discretized variable is evaluated will be indicated with a subscript, *e.g.*, $\boldsymbol{g}_{n+1}$ represents the approximation of $\boldsymbol{g}(t_{n+1})$. Also the discrete approximation of $\boldsymbol{W}(t_n;t_{n+1})$ (resp. $\boldsymbol{U}(t_n;t_{n+1})$, $\boldsymbol{\Lambda}(t_n;t_{n+1})$ or $\boldsymbol{\nu}(t_n;t_{n+1})$) will be denoted as $\boldsymbol{W}_{n+1}$ (resp., $\boldsymbol{U}_{n+1}$, $\boldsymbol{\Lambda}_{n+1}$ or $\boldsymbol{\nu}_{n+1}$).

Firstly, a discrete expression for Eq. (9h) is obtained by time integration

$$\int_{(t_n,t_{n+1}]} \boldsymbol{M}(\boldsymbol{q})\,\mathrm{d}\boldsymbol{w} - \int_{(t_n,t_{n+1}]} \boldsymbol{g}_{\boldsymbol{q}}^T\,[\mathrm{d}\boldsymbol{i} - \tilde{\boldsymbol{\lambda}}\,\mathrm{d}t] = \int_{(t_n,t_{n+1}]} \boldsymbol{f}^*(\boldsymbol{q},\boldsymbol{v},\tilde{\boldsymbol{q}},\tilde{\boldsymbol{v}},\dot{\tilde{\boldsymbol{v}}},t)\,\mathrm{d}t \tag{17}$$

The three terms involved in the above expression can be interpreted as *nonsmooth contributions*, in the sense that each of them represents a difference between the actual motion and the smooth motion. Indeed, $\mathrm{d}\boldsymbol{w}$ is the difference between $\mathrm{d}\boldsymbol{v}$ and $\dot{\tilde{\boldsymbol{v}}}\,\mathrm{d}t$ and the expression of $\boldsymbol{f}^*$ in (8) involves the difference between three operators ($\boldsymbol{f}$, $\boldsymbol{g}_{\boldsymbol{q}}$ and $\boldsymbol{M}$) evaluated for the actual motion and for the smooth motion. According to the NSGA method, the integration of these nonsmooth contributions is based on first-order approximations $\boldsymbol{M}(\boldsymbol{q}(\tau)) = \boldsymbol{M}(\boldsymbol{q}(t)) + \mathcal{O}(h)$

and $\boldsymbol{g}_{\boldsymbol{q}}^T(\boldsymbol{q}(\tau)) = \boldsymbol{g}_{\boldsymbol{q}}^T(\boldsymbol{q}(t)) + \mathcal{O}(h)$, $\forall \tau \in (t_n, t_{n+1}]$ and an Euler implicit discretization

$$\int_{(t_n,t_{n+1}]} \boldsymbol{M}(\boldsymbol{q}(t))\,\mathrm{d}\boldsymbol{w} \simeq \boldsymbol{M}(\boldsymbol{q}_{n+1}) \int_{(t_n,t_{n+1}]} \mathrm{d}\boldsymbol{w} \simeq \boldsymbol{M}(\boldsymbol{q}_{n+1})\,\boldsymbol{W}_{n+1} \tag{18}$$

$$\int_{(t_n,t_{n+1}]} \boldsymbol{g}_{\boldsymbol{q}}^T(\boldsymbol{q}(t))\,[\mathrm{d}\boldsymbol{i} - \tilde{\boldsymbol{\lambda}}\,\mathrm{d}t] \simeq \boldsymbol{g}_{\boldsymbol{q},n+1}^T \int_{(t_n,t_{n+1}]} [\mathrm{d}\boldsymbol{i} - \tilde{\boldsymbol{\lambda}}\,\mathrm{d}t] \simeq \boldsymbol{g}_{\boldsymbol{q},n+1}^T\,\boldsymbol{\Lambda}_{n+1} \tag{19}$$

$$\int_{(t_n,t_{n+1}]} \boldsymbol{f}^*(\boldsymbol{q}(t), \boldsymbol{v}(t), \tilde{\boldsymbol{q}}(t), \tilde{\boldsymbol{v}}(t), \dot{\tilde{\boldsymbol{v}}}(t), t)\,\mathrm{d}t \simeq h\boldsymbol{f}_{n+1}^* \tag{20}$$

This leads to the discrete equation

$$\boldsymbol{M}(\boldsymbol{q}_{n+1})\,\boldsymbol{W}_{n+1} - \boldsymbol{g}_{\boldsymbol{q},n+1}^T\,\boldsymbol{\Lambda}_{n+1} - h\boldsymbol{f}_{n+1}^* = \boldsymbol{0} \tag{21}$$

Secondly, a discrete expression for Eq. (9e) is obtained by time integration

$$\int_{(t_n,t_{n+1}]} \boldsymbol{M}(\boldsymbol{q})\,\dot{\boldsymbol{q}}\,\mathrm{d}t - \int_{(t_n,t_{n+1}]} \boldsymbol{g}_{\boldsymbol{q}}^T\,\boldsymbol{\mu}\,\mathrm{d}t = \int_{(t_n,t_{n+1}]} \boldsymbol{M}(\boldsymbol{q})\,\boldsymbol{v}\,\mathrm{d}t \tag{22}$$

Then, Eq. (14a) is used together with the approximation

$$\boldsymbol{M}(\boldsymbol{q}(t))\,\boldsymbol{W}(t_n;t) \simeq \boldsymbol{g}_{\boldsymbol{q}}^T(\boldsymbol{q}(t))\,\boldsymbol{\Lambda}(t_n;t) + h\boldsymbol{f}^*(\boldsymbol{q}(t), \boldsymbol{v}(t), \tilde{\boldsymbol{q}}(t), \tilde{\boldsymbol{v}}(t), \dot{\tilde{\boldsymbol{v}}}(t), t) \tag{23}$$

which can be derived in a similar way as Eq. (21). We obtain

$$\begin{aligned}\int_{(t_n,t_{n+1}]} \boldsymbol{M}(\boldsymbol{q})\,(\dot{\boldsymbol{q}} - \tilde{\boldsymbol{v}})\,\mathrm{d}t - \int_{(t_n,t_{n+1}]} \boldsymbol{g}_{\boldsymbol{q}}^T\,(\boldsymbol{\Lambda}(t_n;t) + \boldsymbol{\mu}(t))\,\mathrm{d}t \\ = h\int_{(t_n,t_{n+1}]} \boldsymbol{f}^*(\boldsymbol{q}, \boldsymbol{v}, \tilde{\boldsymbol{q}}, \tilde{\boldsymbol{v}}, \dot{\tilde{\boldsymbol{v}}}, t)\,\mathrm{d}t\end{aligned} \tag{24}$$

Again, the three terms involved in the above equality are interpreted as *nonsmooth contributions* and are thus integrated based on first-order approximations and an Euler implicit discretization

$$\begin{aligned}\int_{(t_n,t_{n+1}]} \boldsymbol{M}(\boldsymbol{q})\,(\dot{\boldsymbol{q}} - \tilde{\boldsymbol{v}})\,\mathrm{d}t &\simeq \boldsymbol{M}(\boldsymbol{q}_{n+1}) \int_{(t_n,t_{n+1}]} (\dot{\boldsymbol{q}} - \tilde{\boldsymbol{v}})\,\mathrm{d}t \\ &\simeq \boldsymbol{M}(\boldsymbol{q}_{n+1})\,\boldsymbol{U}_{n+1}\end{aligned} \tag{25}$$

$$\begin{aligned}\int_{(t_n,t_{n+1}]} \boldsymbol{g}_{\boldsymbol{q}}^T\,(\boldsymbol{\Lambda}(t_n;t) + \boldsymbol{\mu}(t))\,\mathrm{d}t &\simeq \boldsymbol{g}_{\boldsymbol{q},n+1}^T \int_{(t_n,t_{n+1}]} (\boldsymbol{\Lambda}(t_n;t) + \boldsymbol{\mu}(t))\,\mathrm{d}t \\ &\simeq \boldsymbol{g}_{\boldsymbol{q},n+1}^T \boldsymbol{\nu}_{n+1}\end{aligned} \tag{26}$$

and Eq. (20). This leads to the discrete equation

$$\boldsymbol{M}(\boldsymbol{q}_{n+1})\,\boldsymbol{U}_{n+1} - \boldsymbol{g}_{\boldsymbol{q},n+1}^T \boldsymbol{\nu}_{n+1} - h^2\boldsymbol{f}_{n+1}^* = \boldsymbol{0} \tag{27}$$

We will see later that this equation shall be used to compute the position correction $\boldsymbol{U}_{n+1}$, after the evaluation of the smooth motion $\tilde{\boldsymbol{q}}_{n+1}$, $\tilde{\boldsymbol{v}}_{n+1}$, $\dot{\tilde{\boldsymbol{v}}}_{n+1}$ but before the evaluation of the velocity jump $\boldsymbol{W}_{n+1}$ and of the total velocity $\boldsymbol{v}_{n+1}$. The dependency of the operator $\boldsymbol{f}^*$ on the velocity then leads to a coupling between the equation for the position correction and the equation for the velocity jump. The mass matrix depends continuously on the configuration $\boldsymbol{q}$. Since $\boldsymbol{q} = \tilde{\boldsymbol{q}} + \boldsymbol{U}$ and $\boldsymbol{U}$ is $\mathcal{O}(h)$, then $\boldsymbol{M}(\boldsymbol{q}) = \boldsymbol{M}(\tilde{\boldsymbol{q}}) + \mathcal{O}(h)$. Therefore, a simplified version

of Eq. (27) is considered to avoid the coupling between the position and velocity variables and to eliminate the dependency of the mass matrix on $\boldsymbol{U}$

$$\boldsymbol{M}(\tilde{\boldsymbol{q}}_{n+1})\,\boldsymbol{U}_{n+1} - \boldsymbol{g}_{\boldsymbol{q},n+1}^{T}\boldsymbol{\nu}_{n+1} - h^2\boldsymbol{f}_{n+1}^{p} = \boldsymbol{0} \tag{28}$$

with

$$\boldsymbol{f}_{n+1}^{p} = \boldsymbol{f}(\boldsymbol{q}_{n+1}, \tilde{\boldsymbol{v}}_{n+1}, t_{n+1}) - \boldsymbol{f}(\tilde{\boldsymbol{q}}_{n+1}, \tilde{\boldsymbol{v}}_{n+1}, t_{n+1}) + (\boldsymbol{g}_{\boldsymbol{q},n+1}^{T} - \boldsymbol{g}_{\tilde{\boldsymbol{q}},n+1}^{T})\tilde{\boldsymbol{\lambda}}_{n+1} \tag{29}$$

In summary, we get the following discrete equations

$$\begin{aligned} \boldsymbol{M}(\boldsymbol{q}_{n+1})\,\boldsymbol{W}_{n+1} - \boldsymbol{g}_{\boldsymbol{q},n+1}^{T}\,\boldsymbol{\Lambda}_{n+1} - h\boldsymbol{f}_{n+1}^{*} &= \boldsymbol{0} && \text{(30a)} \\ \boldsymbol{M}(\tilde{\boldsymbol{q}}_{n+1})\,\boldsymbol{U}_{n+1} - \boldsymbol{g}_{\boldsymbol{q},n+1}^{T}\boldsymbol{\nu}_{n+1} - h^2\boldsymbol{f}_{n+1}^{p} &= \boldsymbol{0} && \text{(30b)} \end{aligned}$$

Let us remark that, if impacts occur and $h \to 0$, $\boldsymbol{W}(t_n; t_{n+1}) = \mathcal{O}(1)$ and $\boldsymbol{\Lambda}(t_n; t_{n+1}) = \mathcal{O}(1)$ since the velocity may exhibit finite jumps, whereas $\boldsymbol{U}(t_n; t_{n+1}) = \mathcal{O}(h)$ and $\boldsymbol{\nu}(t_n; t_{n+1}) = \mathcal{O}(h)$ since the position remain continuous. The position corrections eliminate the introduced drift, because the exact instant at which impacts develop is not known, and the velocity jumps are approximated over the time step.

It should be observed that Eqs. (30) can be further simplified by neglecting the terms which are multiplied by h and h^2, getting the following results:

$$\begin{aligned} \boldsymbol{M}(\boldsymbol{q}_{n+1})\,\boldsymbol{W}_{n+1} - \boldsymbol{g}_{\boldsymbol{q},n+1}^{T}\,\boldsymbol{\Lambda}_{n+1} &= \boldsymbol{0} && \text{(31a)} \\ \boldsymbol{M}(\tilde{\boldsymbol{q}}_{n+1})\,\boldsymbol{U}_{n+1} - \boldsymbol{g}_{\boldsymbol{q},n+1}^{T}\,\boldsymbol{\nu}_{n+1} &= \boldsymbol{0} && \text{(31b)} \end{aligned}$$

By adopting this simplification, the order of the resulting integration algorithm would still be $\mathcal{O}(h)$. However, for some problems it is necessary to use very small stepsizes for obtaining the accuracy of an integrator based on Eqs. (30) as it will be shown in the numerical examples. It can be argued that the force term $\boldsymbol{f}$ is very sensitive to the position correction $\boldsymbol{U}$ for problems involving nonlinear flexible components, and it should be well represented to avoid needing small stepsizes. Therefore, the complete version of the jump equations in Eqs. (30) is retained.

The discrete complementarity conditions at velocity and at position levels are the same as those introduced in [6]:

$$\text{if } g^{j}(\boldsymbol{q}_{n+1}) \leq 0 \text{ then } 0 \leq g_{\boldsymbol{q},n+1}^{j}\,\boldsymbol{v}_{n+1} + e^{j}\,g_{\boldsymbol{q},n}^{j}\,\boldsymbol{v}_{n} \perp \Lambda_{n+1}^{j} \geq 0, \quad \forall j \in \mathcal{U} \tag{32a}$$

$$\boldsymbol{0} \leq \boldsymbol{g}^{\mathcal{U}}(\boldsymbol{q}_{n+1}) \perp \boldsymbol{\nu}_{n+1}^{\mathcal{U}} \geq \boldsymbol{0} \tag{32b}$$

The complete time integration scheme is obtained by combining this first-order approximation of the nonsmooth variables and equations, with a one-step and second-order time integration scheme for the smooth variables. The generalized-α method is used for the smooth part, although other DAE time integration schemes could also be considered. This hybrid time integration scheme is formulated to advance the solution at each step as follows.

3.3.3 Computation of the smooth motion

As previously proposed in Eqs. (9a–9c), the smooth motion is defined by a modified form of the equations of motion at time step $n+1$ where the contributions of the unilateral constraints and associated reaction forces are ignored, i.e.,

$$\begin{aligned} \boldsymbol{M}(\tilde{\boldsymbol{q}}_{n+1})\dot{\tilde{\boldsymbol{v}}}_{n+1} - \boldsymbol{f}(\tilde{\boldsymbol{q}}_{n+1}, \tilde{\boldsymbol{v}}_{n+1}, t_{n+1}) - \boldsymbol{g}_{\tilde{\boldsymbol{q}},n+1}^{\overline{\mathcal{U}},T}\left(k_s\tilde{\boldsymbol{\lambda}}_{n+1}^{\overline{\mathcal{U}}} - p_s\boldsymbol{g}_{\tilde{\boldsymbol{q}},n+1}^{\overline{\mathcal{U}}}\,\tilde{\boldsymbol{v}}_{n+1}\right) &= \boldsymbol{0} \\ -k_s\boldsymbol{g}_{\tilde{\boldsymbol{q}},n+1}^{\overline{\mathcal{U}}}\,\tilde{\boldsymbol{v}}_{n+1} &= \boldsymbol{0} \end{aligned}$$

where the Lagrange multiplier $\tilde{\boldsymbol{\lambda}}^{\overline{\mathcal{U}}}$ has been augmented with a penalty parameter $p_s \geq 0$ in order to add convexity to the objective function [13], and where $k_s > 0$ is a scaling factor for the Lagrange multiplier. The scaling factor k_s contributes to an improvement of the condition number of the iteration matrix yielding a better convergence rate.

These equations are completed with the difference equations:

$$\begin{aligned} \tilde{\boldsymbol{q}}_{n+1} &= \boldsymbol{q}_n + h\boldsymbol{v}_n + h^2(0.5-\beta)\boldsymbol{a}_n + h^2\beta\boldsymbol{a}_{n+1} && \text{(34a)} \\ \tilde{\boldsymbol{v}}_{n+1} &= \boldsymbol{v}_n + h(1-\gamma)\boldsymbol{a}_n + h\gamma\boldsymbol{a}_{n+1} && \text{(34b)} \\ (1-\alpha_m)\boldsymbol{a}_{n+1} + \alpha_m\boldsymbol{a}_n &= (1-\alpha_f)\dot{\tilde{\boldsymbol{v}}}_{n+1} + \alpha_f\dot{\tilde{\boldsymbol{v}}}_n && \text{(34c)} \end{aligned}$$

where $\boldsymbol{a}_{n+1}$ is a pseudo acceleration term that arises in the generalized-α integrator scheme [16]. The numerical coefficients γ, β, α_m, and α_f can be chosen to achieve a desired level of high-frequency dissipation, represented by spectral radius at infinity $\rho_\infty \in [0,1]$, while minimizing unwanted low-frequency dissipation [17]:

$$\alpha_m = \frac{2\rho_\infty - 1}{\rho_\infty + 1}, \quad \alpha_f = \frac{\rho_\infty}{\rho_\infty + 1}, \quad \gamma = 0.5 + \alpha_f - \alpha_m, \quad \beta = 0.25(\gamma + 0.5)^2 \tag{35}$$

Equations (33-34) only involve the smooth position $\tilde{\boldsymbol{q}}_{n+1}$ and velocity $\tilde{\boldsymbol{v}}_{n+1}$ and are thus decoupled from the variables $\boldsymbol{W}_{n+1}$, $\boldsymbol{U}_{n+1}$, $\boldsymbol{q}_{n+1}$, and $\boldsymbol{v}_{n+1}$. Therefore, these five equations can be solved for the five variables $\tilde{\boldsymbol{q}}_{n+1}$, $\tilde{\boldsymbol{v}}_{n+1}$, $\tilde{\boldsymbol{\lambda}}^{\overline{\mathcal{U}}}_{n+1}$, $\dot{\tilde{\boldsymbol{v}}}_{n+1}$ and $\boldsymbol{a}_{n+1}$ using a Newton-Raphson algorithm without any information about the other variables.

3.3.4 Computation of the position correction

After the computation of the smooth motion, the position correction $\boldsymbol{U}_{n+1}$ is computed in order to obtain a position $\boldsymbol{q}_{n+1}$ which satisfies the bilateral constraints $\boldsymbol{g}^{\overline{\mathcal{U}}}(\boldsymbol{q}_{n+1}) = \boldsymbol{0}$ and the non-penetration constraints $\boldsymbol{g}^{\mathcal{U}}(\boldsymbol{q}_{n+1}) \geq \boldsymbol{0}$. Using Eq. (30b) and the discrete complementarity condition (32b), this problem writes

$$\begin{aligned} \boldsymbol{M}(\tilde{\boldsymbol{q}}_{n+1})\boldsymbol{U}_{n+1} - h^2\boldsymbol{f}^p_{n+1} - \boldsymbol{g}^T_{\boldsymbol{q},n+1}\boldsymbol{\nu}_{n+1} &= \boldsymbol{0} \\ -\boldsymbol{g}^{\overline{\mathcal{U}}}(\boldsymbol{q}_{n+1}) &= \boldsymbol{0} \\ \boldsymbol{0} \leq \boldsymbol{g}^{\mathcal{U}}(\boldsymbol{q}_{n+1}) \perp \boldsymbol{\nu}^{\mathcal{U}}_{n+1} &\geq \boldsymbol{0} \end{aligned}$$

An augmented Lagrangian approach as presented by Alart and Curnier [18] is adopted to solve this LCP. Accordingly, the augmented Lagrangian for the set of bilateral and unilateral constraints $\mathcal{C}$ of the sub-problem at position level is given by

$$\begin{aligned} \mathcal{L}^{\mathcal{C}}_p(\boldsymbol{U}_{n+1}, \boldsymbol{\nu}_{n+1}) = &\sum_{j\in\mathcal{U}} \left[-k_p\nu^j_{n+1}g^j_{n+1} + \frac{p_p}{2}g^j_{n+1}g^j_{n+1} - \frac{1}{2p_p}\operatorname{dist}^2(\xi^j_{n+1}, \mathbb{R}^+) \right] \\ &+ \sum_{i\in\overline{\mathcal{U}}} \left[-k_p\nu^i_{n+1}g^i_{n+1} + \frac{p_p}{2}g^i_{n+1}g^i_{n+1} \right] \end{aligned}$$

where $\boldsymbol{\xi}_{n+1} = k_p\boldsymbol{\nu}_{n+1} - p_p\boldsymbol{g}_{n+1}$ is the augmented Lagrange multiplier at position level with $k_p > 0$ a scaling factor and $p_p > 0$ a penalty coefficient, and where the distance of a point $\boldsymbol{z} \in \mathbb{R}^n$ to the convex set C is defined as $\operatorname{dist}(\boldsymbol{z}, C) = ||\boldsymbol{z} - \operatorname{prox}(\boldsymbol{z}, C)||$ with $\operatorname{prox}(\boldsymbol{z}, C) = \operatorname{argmin}_{\boldsymbol{z}^*\in C} \frac{1}{2}||\boldsymbol{z} - \boldsymbol{z}^*||^2$. The adoption of this augmented Lagrangian results in the following

set of equations for the sub-problem at position level:

$$\begin{aligned} \boldsymbol{M}(\tilde{\boldsymbol{q}}_{n+1})\boldsymbol{U}_{n+1} - h^2\boldsymbol{f}^p_{n+1} - \boldsymbol{g}^{\mathcal{A},T}_{\boldsymbol{q},n+1}\,\boldsymbol{\xi}^{\mathcal{A}}_{n+1} &= \boldsymbol{0} && \text{(37a)}\\ -k_p\boldsymbol{g}^{\mathcal{A}}_{n+1} &= \boldsymbol{0} && \text{(37b)}\\ -\frac{k_p^2}{p_p}\boldsymbol{\nu}^{\overline{\mathcal{A}}}_{n+1} &= \boldsymbol{0} && \text{(37c)} \end{aligned}$$

where the active set $\mathcal{A} \equiv \mathcal{A}_{n+1}$ and its complement $\overline{\mathcal{A}} \equiv \overline{\mathcal{A}}_{n+1}$ are given by

$$\begin{aligned} \mathcal{A}_{n+1} &= \overline{\mathcal{U}} \cup \left\{ j \in \mathcal{U} : \xi^j_{n+1} \geq 0 \right\}, && \text{(38a)}\\ \overline{\mathcal{A}}_{n+1} &= \mathcal{C} \setminus \mathcal{A}_{n+1} && \text{(38b)} \end{aligned}$$

The terms associated to the constraints were obtained from the stationary condition $\delta\mathcal{L}^{\mathcal{C}}_p = 0$. The resulting set of equations can be solved in terms of the unknown variables $\boldsymbol{U}_{n+1}$, $\boldsymbol{q}_{n+1}$ and $\boldsymbol{\nu}_{n+1}$ using a Newton semi-smooth method.

3.3.5 Computation of the velocity jump

After the computation of the position field, the velocity jump $\boldsymbol{W}_{n+1}$ is computed such that the velocity $\boldsymbol{v}_{n+1}$ satisfies the bilateral constraints $\boldsymbol{g}^{\overline{\mathcal{U}}}_{\boldsymbol{q}}\,\boldsymbol{v}_{n+1} = \boldsymbol{0}$ and the impact law $g^j_{\boldsymbol{q},n+1}\boldsymbol{v}_{n+1} + e^j_N g^j_{\boldsymbol{q},n}\boldsymbol{v}_n \geq 0$ for all unilateral constraints $j \in \mathcal{U}$ that are active at position level, i.e., that satisfy $\xi^j_{n+1} \geq 0$. Using Eq. (30a) and the discrete complementarity condition (32a), the equations for this problem are given by

$$\begin{aligned} \boldsymbol{M}(\boldsymbol{q}_{n+1})\boldsymbol{W}_{n+1} - h\boldsymbol{f}^*_{n+1} - \boldsymbol{g}^T_{\boldsymbol{q},n+1}\boldsymbol{\Lambda}_{n+1} &= \boldsymbol{0}\\ -\boldsymbol{g}^{\overline{\mathcal{U}}}_{\boldsymbol{q},n+1}\boldsymbol{v}_{n+1} &= \boldsymbol{0}\\ \text{if } g^j(\boldsymbol{q}_{n+1}) \leq 0 \text{ then } 0 \leq g^j_{\boldsymbol{q},n+1}\,\boldsymbol{v}_{n+1} + e^j\,g^j_{\boldsymbol{q},n}\,\boldsymbol{v}_n \perp \Lambda^j_{n+1} &\geq 0, \quad \forall j \in \mathcal{U} \end{aligned}$$

In order to solve this LCP problem, we proceed in a similar manner as we did for the sub-problem at position level. However, the activation of a given unilateral constraint $j \in \mathcal{U}$ at velocity level depends on the activation condition $g^j(\boldsymbol{q}_{n+1}) \leq 0$, see Eq. (39a). This condition is equivalent to $\xi^j_{n+1} \geq 0$, which is more robust from the algorithmic point of view, and therefore adopted in this work. The augmented Lagrange multiplier at velocity level is defined by

$$\boldsymbol{\sigma}_{n+1} = k_v\boldsymbol{\Lambda}_{n+1} - p_v\overset{\circ}{\boldsymbol{g}}_{n+1} \tag{40}$$

where $p_v > 0$ is the penalty parameter, $k_v > 0$ is the scaling factor for the Lagrange multiplier $\boldsymbol{\Lambda}$, and $\overset{\circ}{\boldsymbol{g}}_{n+1}$ is a notation for the impact law

$$\overset{\circ}{g}^j_{n+1} = g^j_{\boldsymbol{q},n+1}\boldsymbol{v}_{n+1} + e^j g^j_{\boldsymbol{q},n}\boldsymbol{v}_n \tag{41}$$

which applies to every $j \in \mathcal{C}$. The coefficients associated to bilateral constraints are trivially defined as $e^j = 0\ \forall\ j \in \mathcal{U}$. Then, the augmented Lagrangian for this case is given by

$$\begin{aligned} \mathcal{L}^{\mathcal{C}}_v(\boldsymbol{W}_{n+1}, \boldsymbol{\Lambda}_{n+1}) = &\sum_{j\in\mathcal{U}} \left[-k_v\Lambda^j_{n+1}\overset{\circ}{g}^j_{n+1} + \frac{p_v}{2}\overset{\circ}{g}^j_{n+1}\overset{\circ}{g}^j_{n+1} - \frac{1}{2p_v}\,\mathrm{dist}^2(\sigma^j_{n+1}, \mathbb{R}^+) \right]\\ &+ \sum_{i\in\overline{\mathcal{U}}} \left[-k_v\Lambda^i_{n+1}\overset{\circ}{g}^i_{n+1} + \frac{p_v}{2}\overset{\circ}{g}^i_{n+1}\overset{\circ}{g}^i_{n+1} \right] \end{aligned}$$

This results in the following set of equations for the problem at velocity level:

$$\begin{aligned} \boldsymbol{M}(\boldsymbol{q}_{n+1})\boldsymbol{W}_{n+1} - h\boldsymbol{f}^{*}_{n+1} - \boldsymbol{g}^{\mathcal{B},T}_{\boldsymbol{q},n+1}\,\boldsymbol{\sigma}^{\mathcal{B}}_{n+1} &= \boldsymbol{0} && \text{(42a)} \\ -k_v \overset{\circ}{\boldsymbol{g}}{}^{\mathcal{B}}_{n+1} &= \boldsymbol{0} && \text{(42b)} \\ -\frac{k_v^2}{p_v}\boldsymbol{\Lambda}^{\overline{\mathcal{B}}}_{n+1} &= \boldsymbol{0} && \text{(42c)} \end{aligned}$$

where the active set $\mathcal{B} \equiv \mathcal{B}_{n+1}$ and its complement $\overline{\mathcal{B}} \equiv \overline{\mathcal{B}}_{n+1}$ are given by

$$\begin{aligned} \mathcal{B}_{n+1} &= \overline{\mathcal{U}} \cup \{j \in \mathcal{A}_{n+1} : \sigma^{j}_{n+1} \geq 0\} && \text{(43a)} \\ \overline{\mathcal{B}}_{n+1} &= \mathcal{C} \setminus \mathcal{B}_{n+1} && \text{(43b)} \end{aligned}$$

The terms associated to the constraints were obtained from the stationary condition $\delta\mathcal{L}^{\mathcal{C}}_{v} = 0$. The resulting set of equations can be solved in terms of the unknown variables $\boldsymbol{W}_{n+1}$, $\boldsymbol{v}_{n+1}$ and $\boldsymbol{\Lambda}_{n+1}$ using a Newton semi-smooth method.

3.3.6 Global numerical procedure

The three sub-problems (33, 37, 42) need to be solved at each time step, one for the smooth motion, another for the position correction, and, lastly, one for the velocity jump. These are computations which can generally be organized in a sequential manner. In [6] and [10], the sub-problem defining the smooth variables involve the total position $\boldsymbol{q}_{n+1}$ and velocity $\boldsymbol{v}_{n+1}$ fields. As a consequence, some global iterations over the three sub-problems have to be implemented, which tend to penalize the numerical cost of the procedure. In contrast, the sub-problem defining the smooth motion is strictly independent of the position correction and velocity jump, and the problem at position level is independent of the velocity jump. In this way, the three sub-problems can be solved in a purely decoupled sequential manner.

Each sub-problem can be solved using a semi-smooth Newton method. The correction terms should satisfy the integration formulae, therefore, the corrections $\Delta\dot{\tilde{\boldsymbol{v}}}_{n+1}$ and $\Delta\boldsymbol{q}_n$ can be eliminated in terms of $\Delta\tilde{\boldsymbol{v}}_{n+1}$, $\Delta\boldsymbol{W}_{n+1}$ and $\Delta\boldsymbol{U}_{n+1}$:

$$\begin{aligned} \Delta\boldsymbol{v}_{n+1} &= \Delta\tilde{\boldsymbol{v}}_{n+1} + \Delta\boldsymbol{W}_{n+1} && \text{(44a)} \\ \Delta\dot{\tilde{\boldsymbol{v}}}_{n+1} &= (1-\alpha_m)/((1-\alpha_f)\gamma h)\,\Delta\tilde{\boldsymbol{v}}_{n+1} && \text{(44b)} \\ \Delta\boldsymbol{q}_{n+1} &= h\,\beta/\gamma\,\Delta\tilde{\boldsymbol{v}}_{n+1} + \Delta\boldsymbol{U}_{n+1} && \text{(44c)} \end{aligned}$$

Then, the vectors of independent corrections are given by

$$\Delta\boldsymbol{x}^{s} = \left\{ \begin{array}{c} \Delta\tilde{\boldsymbol{v}}_{n+1} \\ \Delta\tilde{\boldsymbol{\lambda}}^{\overline{\mathcal{U}}}_{n+1} \end{array} \right\}, \quad \Delta\boldsymbol{x}^{p} = \left\{ \begin{array}{c} \Delta\boldsymbol{U}_{n+1} \\ \Delta\boldsymbol{\nu}^{\mathcal{A}}_{n+1} \\ \Delta\boldsymbol{\nu}^{\overline{\mathcal{A}}}_{n+1} \end{array} \right\}, \quad \Delta\boldsymbol{x}^{v} = \left\{ \begin{array}{c} \Delta\boldsymbol{W}_{n+1} \\ \Delta\boldsymbol{\Lambda}^{\mathcal{B}}_{n+1} \\ \Delta\boldsymbol{\Lambda}^{\overline{\mathcal{B}}}_{n+1} \end{array} \right\} \tag{45}$$

and the correction equations for each sub-problem are obtained as

$$\boldsymbol{S}^{i}_{t}\,\Delta\boldsymbol{x}^{i} = -\boldsymbol{r}^{i}, \quad \text{for} \quad i = s, p, v \tag{46}$$

where $\boldsymbol{r}^{s}$, $\boldsymbol{r}^{p}$ and $\boldsymbol{r}^{v}$ have been used to denote the residuals of Eqs. (33), (37) and (42), respec-

tively, and where the iteration matrices are given by

$$\boldsymbol{S}_t^s = \begin{bmatrix} \boldsymbol{S}_t^{s^*} & -k_s \boldsymbol{g}_{\tilde{\boldsymbol{q}},n+1}^{\overline{\mathcal{U}},T} \\ -k_s \left(\boldsymbol{g}_{\tilde{\boldsymbol{q}},n+1}^{\overline{\mathcal{U}}} + \frac{h\beta}{\gamma} \boldsymbol{G}^s \right) & \boldsymbol{0} \end{bmatrix} \tag{47}$$

$$\boldsymbol{S}_t^p = \begin{bmatrix} \boldsymbol{S}_t^{p^*} & -k_p \boldsymbol{g}_{\boldsymbol{q},n+1}^{\mathcal{A},T} & \boldsymbol{0} \\ -k_p \boldsymbol{g}_{\boldsymbol{q},n+1}^{\mathcal{A}} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & -\frac{k_p^2}{p_p} \boldsymbol{I}^{\overline{\mathcal{A}}} \end{bmatrix} \tag{48}$$

$$\boldsymbol{S}_t^v = \begin{bmatrix} \boldsymbol{S}_t^{v^*} & -k_v \boldsymbol{g}_{\boldsymbol{q},n+1}^{\mathcal{B},T} & \boldsymbol{0} \\ -k_v \boldsymbol{g}_{\boldsymbol{q},n+1}^{\mathcal{B}} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & -\frac{k_v^2}{p_v} \boldsymbol{I}^{\overline{\mathcal{B}}} \end{bmatrix} \tag{49}$$

where $\boldsymbol{I}^{\overline{\mathcal{A}}}$ and $\boldsymbol{I}^{\overline{\mathcal{B}}}$ are identity matrices and

$$\boldsymbol{S}_t^{s^*} = \frac{1-\alpha_m}{h(1-\alpha_f)\gamma} \boldsymbol{M}(\tilde{\boldsymbol{q}}_{n+1}) + \boldsymbol{C}_t + \frac{h\beta}{\gamma} \boldsymbol{K}_t^s,$$

$$\boldsymbol{K}_t^s = \frac{\partial \left(\boldsymbol{M}(\tilde{\boldsymbol{q}}_{n+1}) \dot{\tilde{\boldsymbol{v}}}_{n+1} - \boldsymbol{g}_{\tilde{\boldsymbol{q}},n+1}^{\overline{\mathcal{U}},T} \left(k_s \tilde{\boldsymbol{\lambda}}_{n+1}^{\overline{\mathcal{U}}} - p_s \, \boldsymbol{g}_{\tilde{\boldsymbol{q}},n+1}^{\overline{\mathcal{U}}} \, \tilde{\boldsymbol{v}}_{n+1} \right) \, - \boldsymbol{f}(\tilde{\boldsymbol{q}}_{n+1}, \tilde{\boldsymbol{v}}_{n+1}, t_{n+1}) \right)}{\partial \tilde{\boldsymbol{q}}_{n+1}},$$

$$\boldsymbol{C}_t = p_s \boldsymbol{g}_{\tilde{\boldsymbol{q}},n+1}^{\overline{\mathcal{U}},T} \boldsymbol{g}_{\tilde{\boldsymbol{q}},n+1}^{\overline{\mathcal{U}}} - \frac{\partial \boldsymbol{f}(\tilde{\boldsymbol{q}}_{n+1}, \tilde{\boldsymbol{v}}_{n+1}, t_{n+1})}{\partial \tilde{\boldsymbol{v}}_{n+1}}, \quad \boldsymbol{G}^s = \frac{\partial (\boldsymbol{g}_{\tilde{\boldsymbol{q}},n+1} \tilde{\boldsymbol{v}}_{n+1})}{\partial \tilde{\boldsymbol{q}}_{n+1}}, \tag{50}$$

$$\boldsymbol{S}_t^{p^*} = \boldsymbol{M}(\tilde{\boldsymbol{q}}_{n+1}) - \frac{\partial \left(\boldsymbol{g}_{\boldsymbol{q},n+1}^{\mathcal{A},T} \boldsymbol{\xi}_{n+1}^{\mathcal{A}} \right)}{\partial \boldsymbol{q}_{n+1}} - h^2 \frac{\partial \boldsymbol{f}^p(\boldsymbol{q}_{n+1}, \tilde{\boldsymbol{q}}_{n+1}, \tilde{\boldsymbol{v}}_{n+1}, t_{n+1})}{\partial \boldsymbol{q}_{n+1}},$$

$$\boldsymbol{S}_t^{v^*} = \boldsymbol{M}(\boldsymbol{q}_{n+1}) + p_v \boldsymbol{g}_{\boldsymbol{q},n+1}^{\overline{\mathcal{B}},T} \boldsymbol{g}_{\boldsymbol{q},n+1}^{\overline{\mathcal{B}}} - h \frac{\partial \boldsymbol{f}(\boldsymbol{q}_{n+1}, \boldsymbol{v}_{n+1}, t_{n+1})}{\partial \boldsymbol{v}_{n+1}}$$

As mentioned before, the solution does not depend on the value of parameters k_s, k_p, k_v, p_s, p_p and p_v. Nevertheless, the matrix conditioning and convergence rate do depend on their values, and therefore they are determined as in [19]:

$$k_s = p_s = \frac{\bar{m}}{h}, \quad k_p = p_p = \bar{m}, \quad k_v = p_v = \bar{m}$$

where $\bar{m}$ is a characteristic mass of the problem.

The integrator is summarized in Algorithm 1. The criterion used for checking the Newton scheme convergence in each sub-problem is denoted simply as $||\boldsymbol{r}^i|| < \text{tol}$, for $i = s, p, v$. However, the actual expression of the convergence criterion is given by

$$||\boldsymbol{r}^i|| < \text{tol}_r \left(\sum_k ||\boldsymbol{r}_k^i|| + \text{tol}_f \right) \tag{51}$$

where tol_r is a given relative tolerance, $\boldsymbol{r}_k^i$ is the k-th term contributing to the residual $\boldsymbol{r}^i$, tol_f is a reference value of tolerance and $||\cdot||$ is the L^2 norm of $\cdot$.

Algorithm 1 Decoupled nonsmooth generalized-α time integration scheme

Inputs: initial values $\boldsymbol{q}_0$ and $\boldsymbol{v}_0$;
Compute consistent value of $\dot{\widetilde{\boldsymbol{v}}}_0$
$\boldsymbol{a}_0 := \dot{\widetilde{\boldsymbol{v}}}_0$
for $n = 0$ to $n_{\text{final}} - 1$ **do**
  $\dot{\widetilde{\boldsymbol{v}}}_{n+1} := \boldsymbol{0}$, $\widetilde{\boldsymbol{\lambda}}^{\overline{U}}_{n+1} := \boldsymbol{0}$, $\boldsymbol{\nu}_{n+1} := \boldsymbol{0}$
  $\boldsymbol{\Lambda}_{n+1} := \boldsymbol{0}$, $\boldsymbol{U}_{n+1} := \boldsymbol{0}$, $\boldsymbol{W}_{n+1} := \boldsymbol{0}$
  $\boldsymbol{a}_{n+1} := 1/(1-\alpha_m)(\alpha_f \dot{\widetilde{\boldsymbol{v}}}_n - \alpha_m \boldsymbol{a}_n)$
  $\boldsymbol{v}_{n+1} := \widetilde{\boldsymbol{v}}_{n+1} := \boldsymbol{v}_n + h(1-\gamma)\boldsymbol{a}_n + h\gamma \boldsymbol{a}_{n+1}$
  $\boldsymbol{q}_{n+1} := \boldsymbol{q}_n + h\boldsymbol{v}_n + h^2(1/2\ - \beta)\boldsymbol{a}_n + h^2\beta \boldsymbol{a}_{n+1}$
  Step 1 (smooth motion):
  for $i = 1$ to i_{max} **do**
    Compute residual $\boldsymbol{r}^s$
    if $||\boldsymbol{r}^s|| <$ tol **then** break **end if**
    Compute the iteration matrix $\boldsymbol{S}^s_t$
    $\Delta \boldsymbol{x}^s := -(\boldsymbol{S}^s_t)^{-1}\boldsymbol{r}^s$
    $\tilde{\boldsymbol{v}}_{n+1} := \tilde{\boldsymbol{v}}_{n+1} + \Delta \tilde{\boldsymbol{v}}$
    $\dot{\tilde{\boldsymbol{v}}}_{n+1} := \dot{\tilde{\boldsymbol{v}}}_{n+1} + (1-\alpha_m)/((1-\alpha_f)\gamma h)\Delta \tilde{\boldsymbol{v}}$
    $\boldsymbol{q}_{n+1} := \boldsymbol{q}_{n+1} + h\beta/\gamma \Delta \tilde{\boldsymbol{v}}$
    $\widetilde{\boldsymbol{\lambda}}^{\overline{U}}_{n+1} := \widetilde{\boldsymbol{\lambda}}^{\overline{U}}_{n+1} + \Delta \widetilde{\boldsymbol{\lambda}}^{\overline{U}}$
  end for
  Step 2 (projection on position constraints):
  for $i = 1$ to i_{max} **do**
    Compute residual $\boldsymbol{r}^p$
    if $||\boldsymbol{r}^p|| <$ tol **then** break **end if**
    Compute $\boldsymbol{S}^p_t$
    $\Delta \boldsymbol{x}^p := -(\boldsymbol{S}^p_t)^{-1}\boldsymbol{r}^p$
    $\boldsymbol{U}_{n+1} := \boldsymbol{U}_{n+1} + \Delta \boldsymbol{U}$
    $\boldsymbol{q}_{n+1} := \boldsymbol{q}_{n+1} + \Delta \boldsymbol{U}$
    $\boldsymbol{\nu}_{n+1} := \boldsymbol{\nu}_{n+1} + \Delta \boldsymbol{\nu}$
  end for
  Step 3 (projection on velocity constraints):
  for $i = 1$ to i_{max} **do**
    Compute residual $\boldsymbol{r}^v$
    if $||\boldsymbol{r}^v|| <$ tol **then** break **end if**
    Compute $\boldsymbol{S}^v_t$
    $\Delta \boldsymbol{x}^v := -(\boldsymbol{S}^v_t)^{-1}\boldsymbol{r}^v$
    $\boldsymbol{W}_{n+1} := \boldsymbol{W}_{n+1} + \Delta \boldsymbol{W}$
    $\boldsymbol{v}_{n+1} := \tilde{\boldsymbol{v}}_{n+1} + \boldsymbol{W}_{n+1}$
    $\boldsymbol{\Lambda}_{n+1} := \boldsymbol{\Lambda}_{n+1} + \Delta \boldsymbol{\Lambda}$
  end for
  $\boldsymbol{a}_{n+1} := \boldsymbol{a}_{n+1} + (1-\alpha_f)/(1-\alpha_m)\dot{\tilde{\boldsymbol{v}}}_{n+1}$
end for

4 Numerical examples

Four numerical examples are considered to study the accuracy and robustness of the proposed methodology, Fig. 1. Special emphasis is made on showing the improvements brought to the original NSGA [6]. The average and the maximum number of iterations per time step as well as the rate of convergence with each integrator, are compared. In the case of the original NSGA, the reported number of iterations is actually computed as the number of times that a linearized system of equations has to be solved per global time step. The results obtained with the original NSGA are denoted by CS (Coupled-Solution), whilst the ones resulting from the new algorithm are denoted by DS (Decoupled-Solution). The convergence rates are computed with an error evaluated with the L^1 norm:

$$\text{Error}(h) = \frac{\sum_{n=0}^{N} |f_n - f(t_n)|}{\sum_{n=0}^{N} |f(t_n)|} \tag{52}$$

where N is the number of time steps, f_n is the numerical solution and $f(t_n)$ is the reference solution. The reference solution is taken as the numerical solution for a very small stepsize using the original NSGA. The tolerances tol_r and tol_f in the convergence criterion of the Newton solver, are both adopted equal to 10^{-5}. The spectral radius is taken as $\rho_\infty = 0.8$.

Three of the four examples involve a spatial discretization. When performing the convergence study for these examples, the number of elements is kept constant and only the time stepsize is varied.

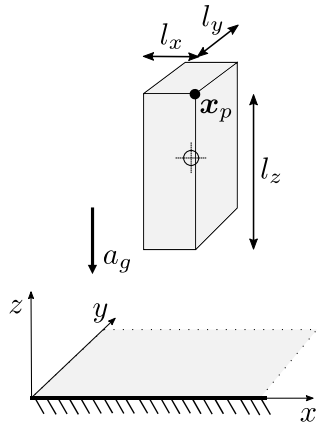
Both integrators have been implemented in the finite element code Oofelie [20]. The finite elements for flexible multibody systems are described in [13] and the coordinates are the nodal absolute translations and rotations. The discretization of rotation variables relies on a Lie group time integration method directly adapted from [21].

The examples have been chosen to highlight important aspects of the proposed strategy. In the first example, the impact of a rigid rectangular parallelepiped body with a large initial angular velocity is solved to show how the presence of gyroscopic forces and of the coupling stemming from nonlinear bilateral constraints (which represent the rigid body) influence the solvers performance. Next, the horizontal impact of an elastic bar is considered to study the ability of the proposed algorithm to deal with flexibility. In the third example, the behaviour of the integrators for problems with nonlinear flexible beams is considered by studying the bouncing of a flexible pendulum. Lastly, the bouncing of a 3D flexible cube is examined to investigate the performance of the method for 3D nonlinear finite element models.

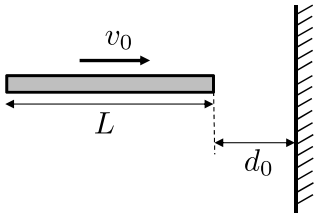
4.1 Impact of a rigid rectangular parallelepiped body

This example consists in the impact of a rigid rectangular parallelepiped body on a rigid support, Fig. 1(a). This problem does not involve flexibility, but is subject to nonlinear bilateral constraints that model the rigidity of the body. The nonlinear bilateral constraints couple the smooth problem to the sub-problem at position level in the CS algorithm, influencing the required number of iterations for convergence. A large initial angular velocity is imposed, making it necessary to accurately account for gyroscopic effects. The important contribution of gyroscopic forces allows to evaluate the appropriateness of the approximation $\boldsymbol{f}(\boldsymbol{q}_{n+1}, \boldsymbol{v}_{n+1}, t_{n+1}) \approx \boldsymbol{f}(\boldsymbol{q}_{n+1}, \tilde{\boldsymbol{v}}_{n+1}, t_{n+1})$ at the position level in the DS solver.

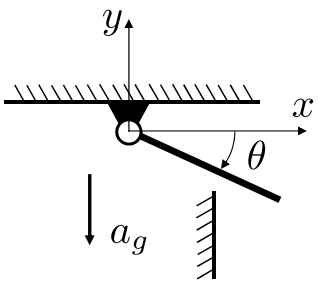
The parameters of the problem are $m = 46.8$ kg, $l_x = 0.1$ m, $l_y = 0.2$ m, $l_z = 0.3$ m, $a_g = 9.81$ m/s^2 and restitution coefficient $e = 0.8$. The inertia tensor $\boldsymbol{J}$ is diagonal with entries $J_{xx} = m(l_y^2 + l_z^2)/12$, $J_{yy} = m(l_x^2 + l_z^2)/12$ and $J_{zz} = m(l_y^2 + l_x^2)/12$. The body center



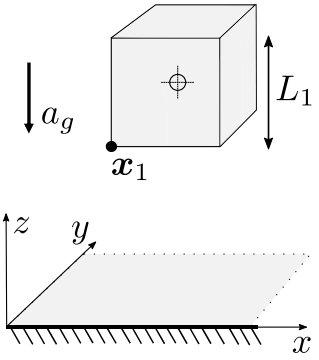
(a) Impact of a rigid rectangular parallelepiped.



(b) Horizontal impact of an elastic bar.



(c) Bouncing flexible pendulum.



(d) Bouncing 3D flexible cube.

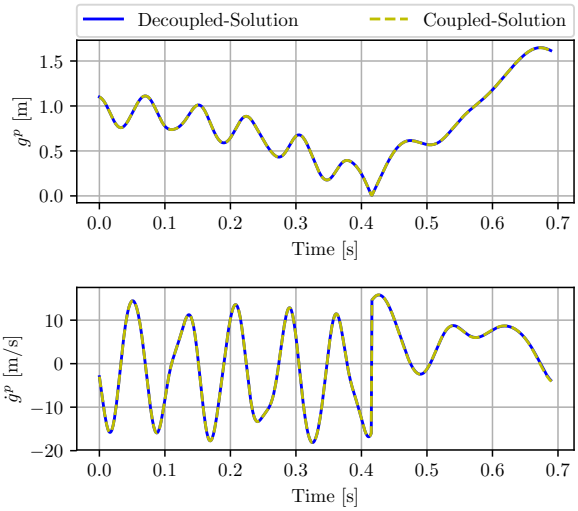
Figure 1: Numerical examples.

of mass has initial conditions: angular velocity $\boldsymbol{\Omega}_0 = [60, 60, 0]^T$ rad/s, translation velocity $\boldsymbol{v}_0 = [3, 0, 0]^T$ m/s and initial position $\boldsymbol{x}_0 = [0, 0, 1]^T$ m. The reference solution is obtained with a stepsize of $h = 10^{-5}$ s using the CS solver. The z-displacement and velocity of node $\boldsymbol{x}_p$ for a stepsize of $h = 10^{-3}$ s is plotted in Fig. 2(a), whilst in Fig. 2(b) the convergence analysis of both integrators is shown. As it can be seen, both methods converge with the same convergence rate and error. However, the average and the maximum number of iterations, Figs. 3(a-b), make clear that the new splitting strategy of DS is more robust than CS for problems with nonlinear bilateral constraints and important gyroscopic contributions. In addition, we observe that the approximation $\boldsymbol{f}(\boldsymbol{q}_{n+1}, \boldsymbol{v}_{n+1}, t_{n+1}) \approx \boldsymbol{f}(\boldsymbol{q}_{n+1}, \tilde{\boldsymbol{v}}_{n+1}, t_{n+1})$ at position level in the DS solver does not affect the accuracy of the obtained results for the considered example.

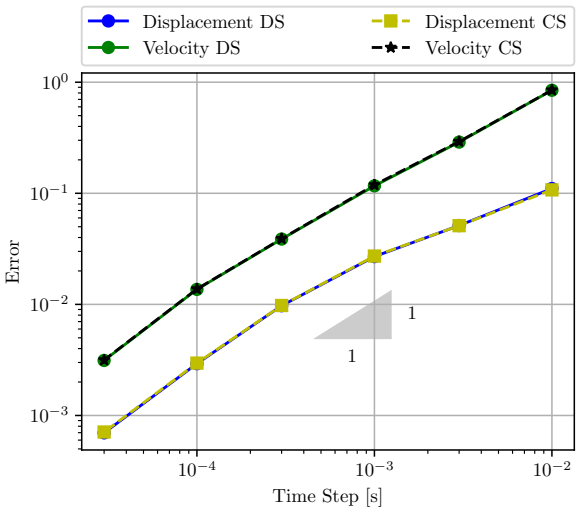
4.2 Horizontal impact of an elastic bar

The horizontal impact of an elastic bar is analyzed, see Fig. 1(b). This example deals with a linear flexible model with no bilateral constraints, allowing to understand convergence problems associated to flexibility. It should be noted that in the case of the CS solver, the smooth sub-problem is coupled to the position sub-problem through the internal elastic forces which depend on the total position $\boldsymbol{q}$.

The bar starts moving from a distance d_0 in an undeformed configuration with a uniform

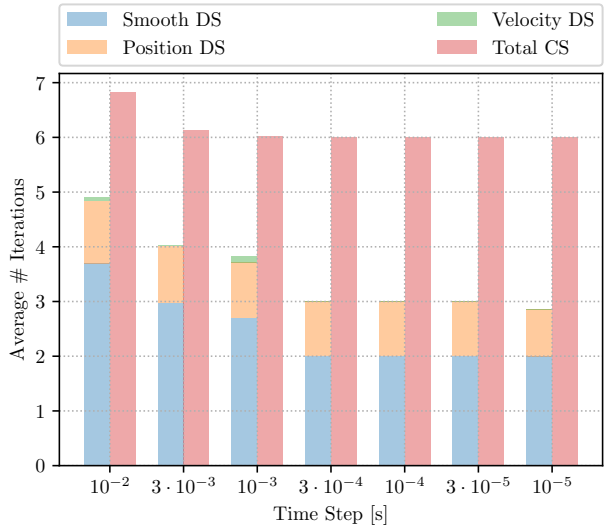


(a) Solution obtained for both solvers for $h = 10^{-3}$ s. Showing the gap and its time derivative for the node $\boldsymbol{x}_p$, see Fig. 1(a).

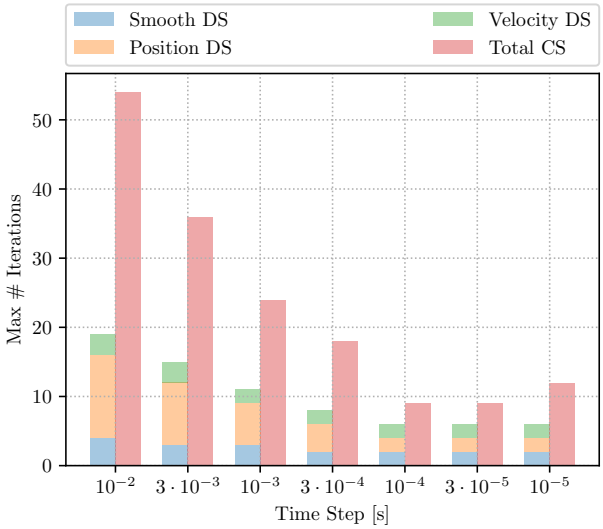


(b) Convergence analysis performed for the time window $t = (0, 0.69)$ s for the z component of the displacement and the velocity fields of node $\boldsymbol{x}_p$.

Figure 2: Results obtained for the impact of a rigid rectangular parallelepiped body.



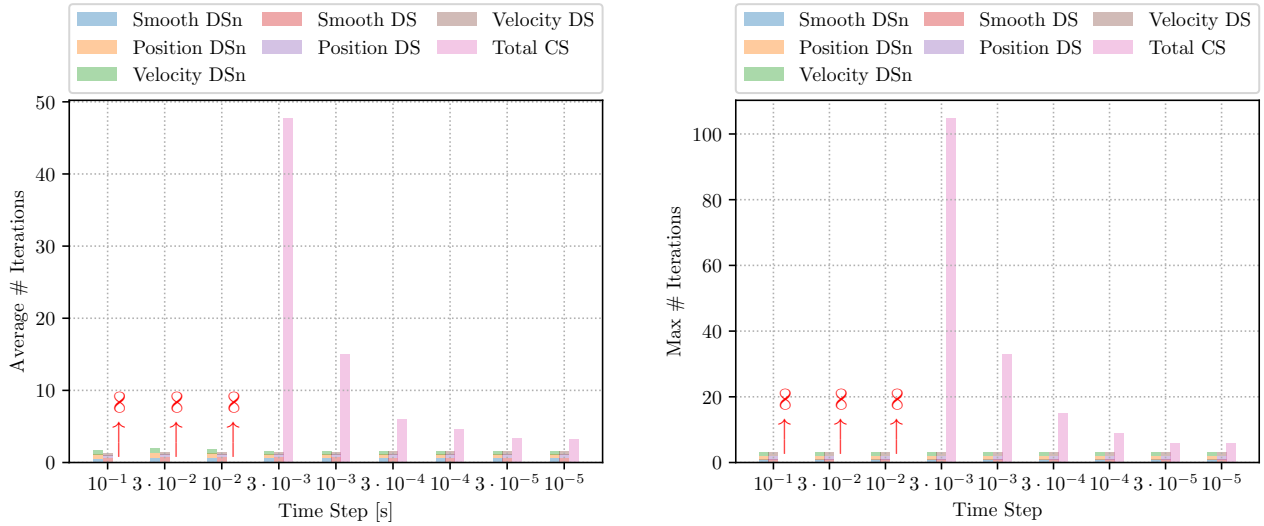
(a) Average number of iterations.



(b) Maximum number of iterations.

Figure 3: Rigid rectangular parallelepiped problem: number of iterations taken by the DS and CS solvers.

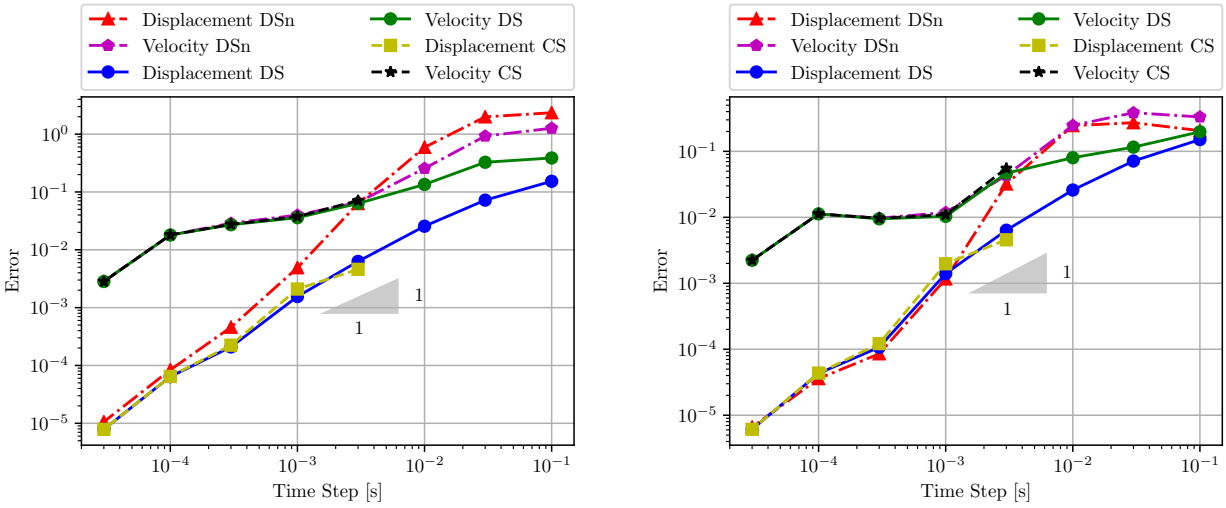
initial velocity field v_0, and bounces back after impacting a rigid wall. The impact is horizontal and no gravity is considered [6]. The parameters of this test are: Young modulus $E = 900$ Pa, density $\rho = 1$ kg/m^3, undeformed initial length $L = 10$ m, initial distance from the wall $d_0 = 5.005$ m, initial velocity $v_0 = 10$ m/s and restitution coefficient $e = 0$. The bar is discretized using 200 finite elements. The reference solution is computed with a time step of $h = 10^{-5}$ s using the CS solver. Three different nonsmooth integrators are tested: the original NSGA (CS) solver, the decoupled NSGA based on Eqs. (30a, 30b), and the decoupled NSGA based on Eqs. (31a, 31b). The solutions obtained with the two latter options are respectively



(a) Average number of iterations.

(b) Maximum number of iterations.

Figure 4: Horizontal impact of an elastic bar: number of iterations for convergence.



(a) Convergence analysis performed for the time window $t = (0, 1.5)$ s for the horizontal displacement and velocity of the node located at 0.45 m from the tip of the bar.

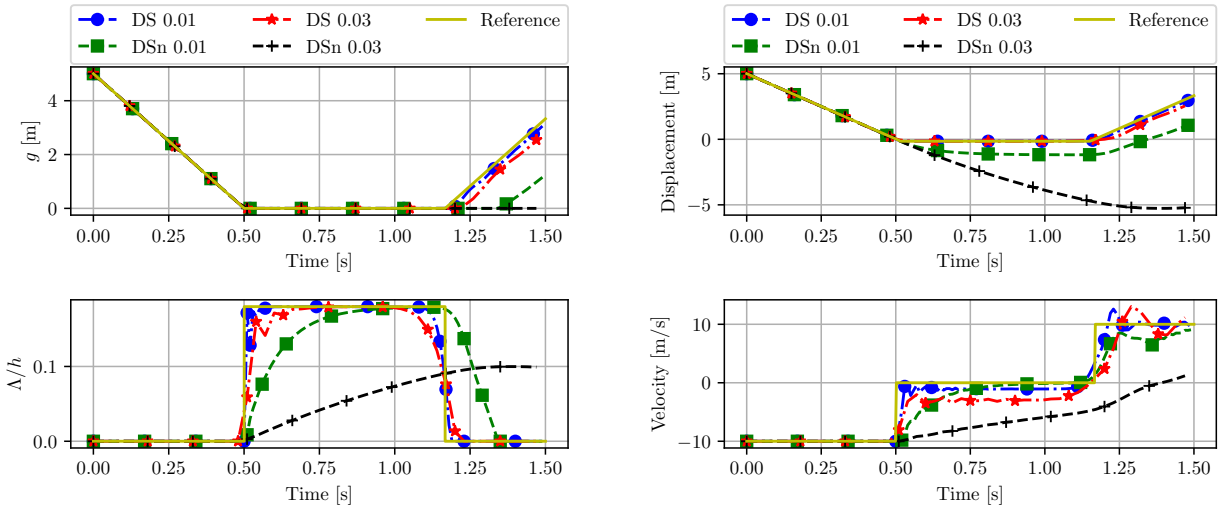
(b) Convergence analysis performed for the time window $t = (0, 1.5)$ s for the horizontal displacement and velocity of the tip node of the bar.

Figure 5: Convergence analysis for the horizontal impact of an elastic bar.

denoted by DS and DSn (Decoupled-Solution neglecting the $\boldsymbol{f}^p$ and $\boldsymbol{f}^*$ terms).

The average and maximum number of iterations for each strategy can be observed in Figs. 4(a-b). Since the problem is linear, the maximum number of iterations in the DS and DSn strategies is only one per sub-problem for any value of stepsize h. On the other hand, the number of iterations required by the original solver (CS) is quite high, and it even diverged for stepsizes greater than or equal to $h = 10^{-2}$ s. Therefore, the advantage of the proposed splitting becomes quite clear.

Figures 5(a-b) show the convergence analysis for two nodes of the bar (one at the tip and



(a) Gap g and contact force Λ/h at the tip of the bar.

(b) Displacement and velocity of the node located at 0.45 m from the tip of the bar.

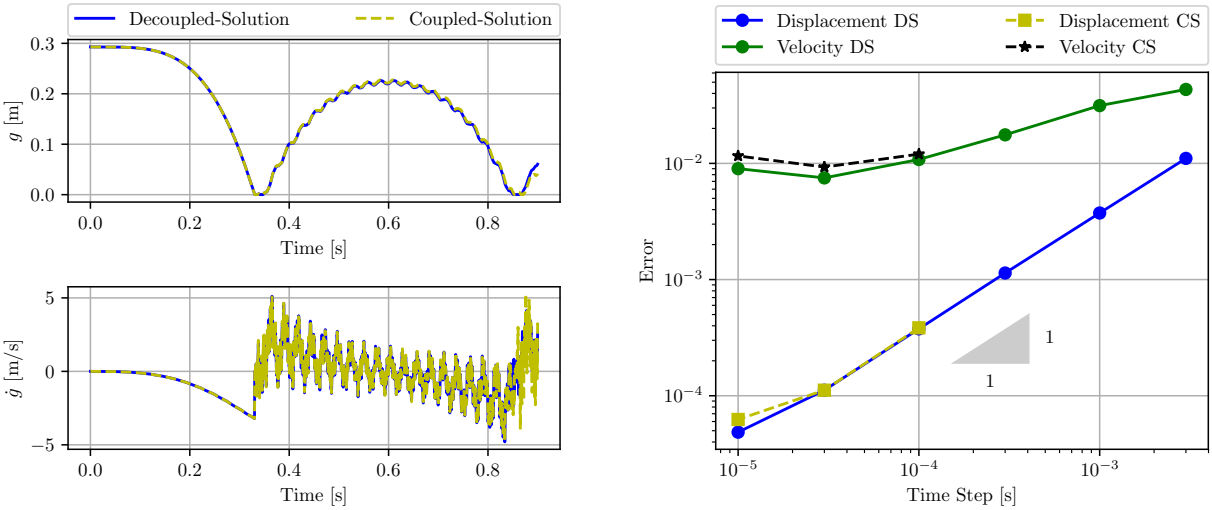
Figure 6: Comparison of the results obtained for different stepsizes using the decoupled solver with (DSn) and without (DS) neglecting the $\boldsymbol{f}^p$ and $\boldsymbol{f}^*$ terms.

another located 0.45 m from the tip). It can be seen that the robustness improvement in the DS strategy is obtained without damaging the convergence rate neither the accuracy. However, in the decoupled strategy DSn, where the $\boldsymbol{f}^p$ and $\boldsymbol{f}^*$ terms were neglected, the error is larger than for the DS strategy for moderately large stepsizes. A comparison of the results of DSn and DS for different stepsizes is shown in Figs. 6(a-b). It is important to highlight that for moderately large stepsizes, the DSn solution does not reproduce certain key aspects of the physical solution, such as, for instance, the bouncing back ot the bar ($h = 0.03$ s), or a quite long delay in the bouncing back with respect to the reference solution ($h = 0.01$ s), Fig. 6(a). In addition, the contact forces deviate considerably from what is expected. Therefore, these results indicate that the term $\boldsymbol{f}(\boldsymbol{q}, \tilde{\boldsymbol{v}}, t_{n+1}) - \boldsymbol{f}(\tilde{\boldsymbol{q}}, \tilde{\boldsymbol{v}}, t_{n+1}) \approx \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{q}} \boldsymbol{U}_{n+1}$ has an important contribution to the position correction for large values of h and cannot be neglected. Another error is evidenced by observing the computed displacement and velocity at the node located at 0.45 m from the tip of the bar, Fig. 6(b). When neglecting the $\boldsymbol{f}^p$ and $\boldsymbol{f}^*$ terms for $h = 0.03$ s, the computed solution satisfies the non-penetration condition at the tip node, Fig. 6(a), but the node at 0.45 m from the tip penetrates the wall, see Fig. 6(b).

4.3 Bouncing of a flexible pendulum

The bouncing of a flexible beam pendulum hitting an obstacle is next studied, Fig. 1(c). This test allows to assess the performance of the integrators for problems involving nonlinear flexible beams. The pendulum is constrained to swing in the x-y plane around a pivot located at the origin. The properties of the beam are: undeformed length $L = 1$ m, cross-section area $A = 10^{-4}$ m^2, cross-section inertia $I = 8.33\ 10^{-10}$ m^4, shear section area $A_s = 5/6A$, Young modulus $E = 2.1\ 10^{11}$ N/m^2, density $\rho = 7800$ kg/m^3 and Poisson ratio $\nu = 0.3$. The beam is in horizontal position at the initial configuration with zero velocity, and begins to fall under the action of gravity $a_g = 9.81$ m/s^2. The tip of the beam hits a rigid wall located at $x = \sqrt{2}/2$, with a coefficient of restitution $e = 0$.

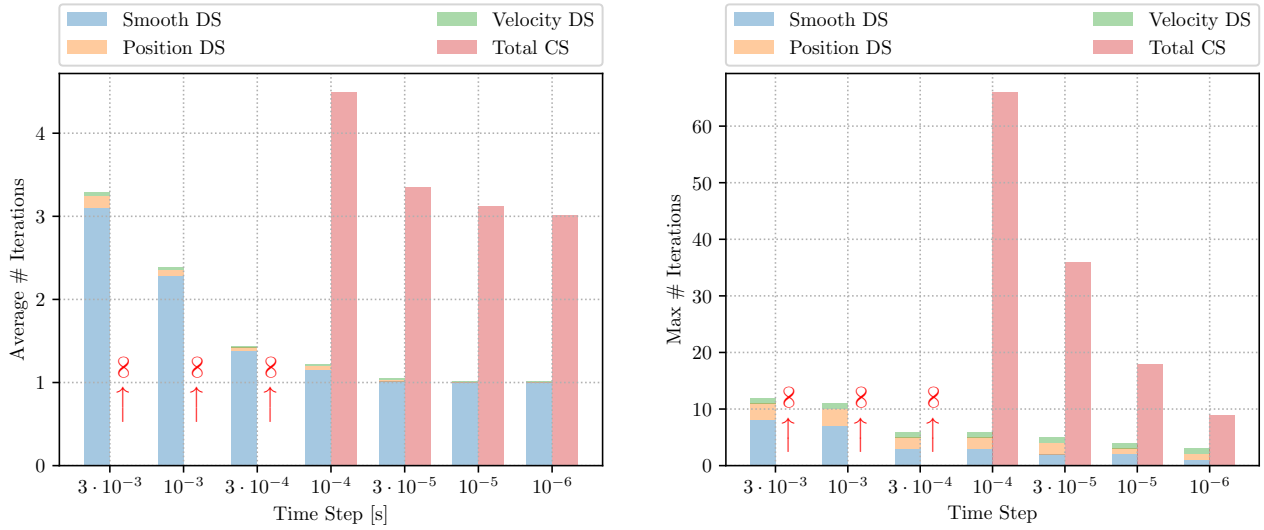
The beam is discretized into 8 equally-spaced nonlinear beam elements. The unilateral constraint representing the impact condition is $0 \leq g^{\mathcal{U}} = x - \sqrt{2}/2 \perp \mathrm{d}i^{\mathcal{U}} \geq 0$ and is enforced at the tip node. The reference solution is computed with a stepsize of $h = 10^{-6}$ s by using the original NSGA (CS) solver.



(a) Solution obtained for both solvers for $h = 10^{-4}$ s. Showing the y component of the displacement and the velocity fields at the tip of the beam.

(b) Convergence analysis performed for the time window $t = (0, 0.35)$ s for the y component of the displacement and the velocity of the tip node of the pendulum.

Figure 7: Results obtained for the bouncing flexible pendulum.



(a) Average number of iterations.

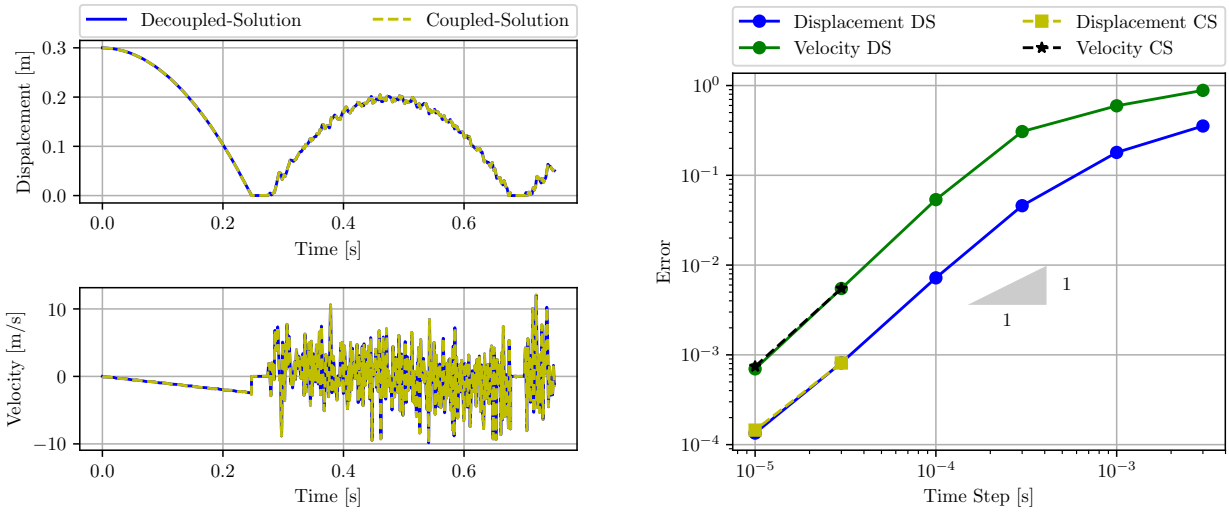
(b) Maximum number of iterations.

Figure 8: Bouncing flexible pendulum: number of iterations taken by the DS and CS solvers.

The y component of the displacement and the velocity at the tip of the beam calculated using $h = 10^{-4}$ s for both solvers are plotted in Fig. 7(a). The average and maximum number of iterations are shown in Figs. 8(a-b). It can be observed that for the three largest stepsizes, the

original NSGA does not converge, Fig. 8. In addition, when it converges for smaller stepsizes, it takes a large number of iterations at impact events (see the maximum number of iterations). On the contrary, the proposed decoupled solver (DS) requires a small number of iterations for convergence.

4.4 Bouncing of a 3D flexible cube



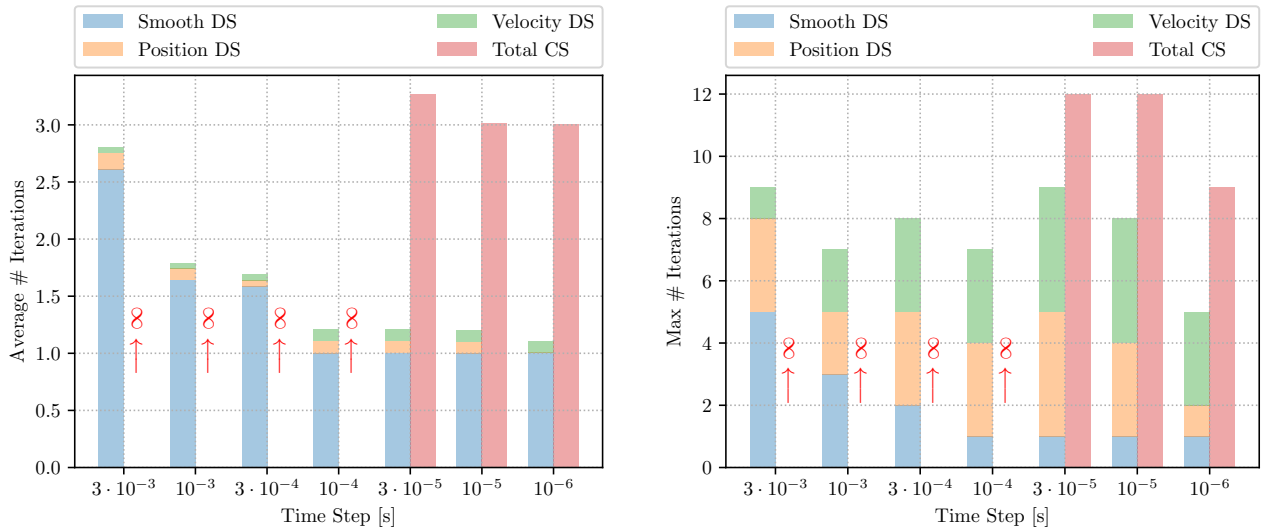
(a) Solution obtained for both solvers for $h = 3{\cdot}10^{-5}$ s. Showing the z component of the displacement and the velocity fields at node $\boldsymbol{x}_1$, see Fig. 1(d).

(b) Convergence analysis performed for the time window $t = (0, 0.3)$ s.

Figure 9: Results obtained for the bouncing of a 3D flexible cube.

In what follows a 3D example is considered. It consists in a 3D flexible cube which bounces against a rigid plane due to the action of gravity $a_g = 9.81$ m/s^2, Fig. 1(d). The cube is discretized into 125 equal tri-linear hexahedral geometrically nonlinear finite elements. It is assumed that the material responds linearly. The cube has an undeformed side length $L_1 = 0.1$ m, the Young modulus is $E = 90$ Pa, the density is $\rho = 1$ kg/m^3, the Poisson ratio is $\nu = 0.3$ and the centroid of the cube is initially located at 0.35 m from the floor with a zero initial velocity. The restitution coefficient is $e = 0$. Contact elements are defined between the floor plane and each node on the cube face.

The z component of the displacement and the velocity at node $\boldsymbol{x}_1$, computed using $h = 3 \cdot 10^{-5}$ s for both solvers, are plotted in Fig. 9(a). The convergence analysis for both solvers is shown in Fig. 9(b). As it can be observed, a convergence rate close to order 1 is achieved for the proposed solver (DS). The original NSGA (CS) algorithm also exhibits order 1 with the same accuracy, but only for stepsizes smaller than or equal to $3 \cdot 10^{-5}$ s. For large stepsizes, the CS solver does not converge (Figs. 10(a-b)). The proposed DS solver deals well with flexible problems characterized by large displacements for any time stepsize. On the contrary, the original NSGA (CS) requires adopting a very small stepsize to get convergence.



(a) Average number of iterations.

(b) Maximum number of iterations.

Figure 10: Bouncing 3D flexible cube: number of iterations taken by the DS and CS solvers.

5 Conclusions

A fully decoupled nonsmooth generalized-α integration method was presented. Like its predecessor, it does not suffer from any drift phenomena as it imposes the constraints both at position and at velocity levels. Additionally, it is well suited for problems with vibration effects as it integrates the smooth component of the motion with the second order accurate generalized-α method.

The algorithm was implemented as a sequence of three sub-problems to be solved at each time step. The most distinctive feature of the new algorithm was that the sub-problem defining the smooth part of the motion is strictly independent of the position correction and of the velocity jump, so that the solution of the three sub-problems could be performed in a purely decoupled sequential manner.

Four numerical examples were presented, showing that the proposed method improves the robustness for problems involving nonlinear bilateral constraints and/or flexible elements, without deteriorating the accuracy of the original NSGA method. The number of iterations was reduced and much larger time steps could be adopted.

A variant of the new method, in which the $\boldsymbol{f}^p$ and $\boldsymbol{f}^*$ terms were neglected, was analyzed in the examples. The computed results showed that neglecting those terms could lead to results of bad quality if sufficiently small stepsizes were not adopted. Hence, it was recommended to take into account those terms in the implementation of the decoupled algorithm. The application of this new algorithm to deal with frictional contact problems is studied in a companion paper.

Acknowledgements

This work has received financial support from Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Agencia Nacional de Promoción Científica y Tecnológica (ANPCyT) PICT2015-1067, Universidad Tecnológica Nacional PID-UTN UTI4790TC and by the M4 project funded by the Walloon Region (Pôle MecaTech), which are gratefully acknowledged.

References

[1] Paoli, L., and Schatzman, M., 2002. "A numerical scheme for impact problems I: The one-dimensional case". *SIAM Journal on Numerical Analysis,* **40**(2), pp. 702–733.

[2] Paoli, L., and Schatzman, M., 2002. "A numerical scheme for impact problems II: The multidimensional case". *SIAM Journal on Numerical Analysis,* **40**(2), pp. 734–768.

[3] Jean, M., and Moreau, J. J., 1987. "Dynamics in the presence of unilateral contacts and dry friction: A numerical approach". In *Unilateral Problems in Structural Analysis — 2.* Springer Vienna, pp. 151–196.

[4] Moreau, J. J., 1988. "Unilateral contact and dry friction in finite freedom dynamics". In *Nonsmooth Mechanics and Applications.* Springer Vienna, pp. 1–82.

[5] Jean, M., 1999. "The non-smooth contact dynamics method". *Computer Methods in Applied Mechanics and Engineering,* **177**(3-4), pp. 235–257.

[6] Brüls, O., Acary, V., and Cardona, A., 2014. "Simultaneous enforcement of constraints at position and velocity levels in the nonsmooth generalized-α scheme". *Computer Methods in Applied Mechanics and Engineering,* **281**, pp. 131 – 161.

[7] Chen, Q., Acary, V., Virlez, G., and Brüls, O., 2013. "A nonsmooth generalized-α scheme for flexible multibody systems with unilateral constraints". *International Journal for Numerical Methods in Engineering,* **96**(8), pp. 487–511.

[8] Schindler, T., and Acary, V., 2014. "Timestepping schemes for nonsmooth dynamics based on discontinuous galerkin methods: Definition and outlook". *Mathematics and Computers in Simulation,* **95**, pp. 180–199.

[9] Schindler, T., Rezaei, S., Kursawe, J., and Acary, V., 2015. "Half-explicit timestepping schemes on velocity level based on time-discontinuous galerkin methods". *Computer Methods in Applied Mechanics and Engineering,* **290**, pp. 250–276.

[10] Brüls, O., Acary, V., and Cardona, A., 2018. "On the constraints formulation in the nonsmooth generalized-α method". In *Advanced Topics in Nonsmooth Dynamics.* Springer International Publishing, pp. 335–374.

[11] Acary, V., 2013. "Projected event-capturing time-stepping schemes for nonsmooth mechanical systems with unilateral contact and coulomb's friction". *Computer Methods in Applied Mechanics and Engineering,* **256**, pp. 224–250.

[12] Schoeder, S., Ulbrich, H., and Schindler, T., 2013. "Discussion of the gear-gupta-leimkuhler method for impacting mechanical systems". *Multibody System Dynamics,* **31**(4), pp. 477–495.

[13] Géradin, M., and Cardona, A., 2001. *Flexible Multibody Dynamics: A Finite Element Approach.* Wiley.

[14] Moreau, J., 1999. "Numerical aspects of the sweeping process". *Computer Methods in Applied Mechanics and Engineering,* **177**(3-4), pp. 329–349.

[15] Gear, C., Leimkuhler, B., and Gupta, G., 1985. "Automatic integration of euler-lagrange equations with constraints". *Journal of Computational and Applied Mathematics,* **12-13**, pp. 77–90.

[16] Arnold, M., and Brüls, O., 2007. "Convergence of the generalized-α scheme for constrained mechanical systems". *Multibody System Dynamics,* **18**(2), pp. 185–202.

[17] Chung, J., and Hulbert, G. M., 1993. "A time integration algorithm for structural dynamics with improved numerical dissipation: The generalized-α method". *Journal of Applied Mechanics,* **60**(2), pp. 371–375.

[18] Alart, P., and Curnier, A., 1991. "A mixed formulation for frictional contact problems prone to newton like solution methods". *Computer Methods in Applied Mechanics and Engineering,* **92**(3), pp. 353 – 375.

[19] Cardona, A., and Géradin, M., 1994. "Numerical integration of second order differential—algebraic systems in flexible mechanism dynamics". In *Computer-Aided Analysis of Rigid and Flexible Mechanical Systems*. Springer Netherlands, pp. 501–529.

[20] Cardona, A., Klapka, I., and Géradin, M., 1994. "Design of a new finite element programming environment". *Engineering Computations,* **11**, pp. 365–381.

[21] Brüls, O., Cardona, A., and Arnold, M., 2012. "Lie group generalized-α time integration of constrained flexible multibody systems". *Mechanism and Machine Theory,* **48**, pp. 121–137.